The Limits of Efficiency for Open- and Closed-World Query Evaluation Under Guarded TGDs

The Limits of Efficiency for Open- and Closed-World Query Evaluation Under Guarded TGDs

Pablo Barceló
IMC, PUC Chile & IMFD Chile
pbarcelo@ing.puc.cl

Victor Dalmau
Universitat Pompeu Fabra
victor.dalmau@upf.edu

Cristina Feier
University of Bremen
feier@uni-bremen.de

Carsten Lutz
University of Bremen
clu@uni-bremen.de

Andeas Pieris
University of Edinburgh
apieris@inf.ed.ac.uk

ABSTRACT

Ontology-mediated querying and querying in the presence of constraints are two key database problems where tuple-generating dependencies (TGDs) play a central role. In ontology-mediated querying, TGDs can formalize the ontology and thus derive additional facts from the given data, while in querying in the presence of constraints, they restrict the set of admissible databases. In this work, we study the limits of efficient query evaluation in the context of the above two problems, focussing on guarded and frontier-guarded TGDs and on UCQs as the actual queries. We show that a class of ontology-mediated queries (OMQs) based on guarded TGDs can be evaluated in FPT iff the OMQs in the class are equivalent to OMQs in which the actual query has bounded treewidth, up to some reasonable assumptions. For querying in the presence of constraints, we consider classes of constraint-query specifications (CQSs) that bundle a set of constraints with an actual query. We show a dichotomy result for CQSs based on guarded TGDs that parallels the one for OMQs except that, additionally, FPT coincides with PTime combined complexity. The proof is based on a novel connection between OMQ and CQS evaluation. Using a direct proof, we also show a similar dichotomy result, again up to some reasonable assumptions, for CQSs based on frontier-guarded TGDs with a bounded number of atoms in TGD heads. Our results on CQSs can be viewed as extensions of Grohe's well-known characterization of the tractable classes of CQs (without constraints). Like Grohe's characterization, all the above results assume that the arity of relation symbols is bounded by a constant. We also study the associated meta problems, i.e., whether a given OMQ or CQS is equivalent to one in which the actual query has bounded treewidth.

ACM Reference Format:
Pablo Barceló, Victor Dalmau, Cristina Feier, Carsten Lutz, and Andeas Pieris. 2020. The Limits of Efficiency for Open- and Closed-World Query Evaluation Under Guarded TGDs. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 33 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Tuple-generating dependencies (TGDs) are a prominent rule-based formalism at the core of several areas that are of central importance to databases and artificial intelligence. They are first-order implications of the form $\forall\bar{x}\forall\bar{y}\,(\phi(\bar{x},\bar{y}) \rightarrow \exists\bar{z}\,\psi(\bar{x},\bar{z}))$, where ϕ and ψ are conjunctions of relational atoms, and they essentially state that some tuples (facts) in a relational instance imply the presence of some other tuples in that instance (hence the name "tuple-generating"). In query evaluation over relational databases, TGDs have two facets: (1) they can be used as ontology axioms that allow us to derive additional facts from the given data, which supports more complete query answers, and (2) they can be used as integrity constraints that restrict the set of admissible databases, which paves the way to constraint-aware query optimization. In fact, TGDs were originally introduced as a unifying framework for the large range of relational constraints introduced in the 1970s and the 1980s [1].

The Two Facets of TGDs. When a set of TGDs is used as an ontology, they are often bundled with an actual database query, typically a union of conjunctive queries (UCQ), to form a composite ontology-mediated query (OMQ) [11, 12]. As discussed above, adding the ontology serves the purpose of delivering more complete answers to queries. It also enriches the vocabulary available for querying as it may introduce new relation symbols that are not part of the data signature. An OMQ language is a pair $(\mathbb{C},\mathbb{Q})$, with $\mathbb{C}$ being a class of TGDs and $\mathbb{Q}$ a query language, which collects all the OMQs in which the ontology is formulated in $\mathbb{C}$ and the actual query comes from $\mathbb{Q}$ [12]. The main problem of concern for an OMQ language $(\mathbb{C},\mathbb{Q})$ is OMQ evaluation under an open-world semantics, i.e., we are looking for answers to the actual query from $\mathbb{Q}$ that are logically entailed by the input database and the ontology from $\mathbb{C}$.

On the other hand, when a set of TGDs is used as relational integrity constraints, the TGDs together with the actual query can be bundled together into what we call a constraint-query specification (CQS). A class of CQSs is a pair $(\mathbb{C},\mathbb{Q})$, where, as in OMQs, $\mathbb{C}$ is a class of TGDs and $\mathbb{Q}$ a query language. In this setting, however, the problem of interest is CQS evaluation under a closed-world semantics, i.e., we are looking for answers to the query from $\mathbb{Q}$ over the input database, which is promised to comply with the set of constraints from $\mathbb{C}$. In other words, we directly evaluate the query over the database, which we know to satisfy the set of constraints.

TGDs and Guardedness. It is well-known that OMQ evaluation in $(\mathbb{TGD},\mathbb{UCQ})$, where $\mathbb{TGD}$ is the class of arbitrary TGDs, and $\mathbb{UCQ}$ the class of union of conjunctive queries, is an undecidable

problem; see, e.g., [14]. This has led to an intensive research activity for identifying restrictions on TGDs that ensure the decidability of OMQ evaluation; see, e.g., [3, 14, 16, 29] – the list is by no means exhaustive. One of the most robust syntactic paradigms that emerged from this extensive effort, which is central to our work, is guardedness. A TGD is guarded if the left-hand of the implication, called the body, has an atom that contains (or guards) all the universally quantified variables [14]; let $\mathbb{G}$ be the class of guarded TGDs. A natural generalization is to guard only the variables that appear also in the right-hand side, called the head, which leads to the class of frontier-guarded TGDs $\mathbb{FG}$ [3]. The complexity of OMQ evaluation in $(\mathbb{C}, \mathbb{UCQ})$, where $\mathbb{C} \in \{\mathbb{G}, \mathbb{FG}\}$, is by now well-understood. In both cases, it is 2ExpTime-complete [2, 14], and it remains hard even in the case of bounded-arity schemas, i.e., 2ExpTime-complete in $(\mathbb{FG}, \mathbb{UCQ})$ [2], and ExpTime-complete in $(\mathbb{G}, \mathbb{UCQ})$ [14].

Although guarded and frontier-guarded TGDs have been proposed as ontology languages, they can also naturally serve as classes of integrity constraints [9]. Note, for example, that the important class of referential integrity constraints (or inclusion dependencies) is a very special case of guarded TGDs. Now, the complexity of CQS evaluation in $(\mathbb{C}, \mathbb{UCQ})$, where $\mathbb{C} \in \{\mathbb{G}, \mathbb{FG}\}$, coincides with that of query evaluation for $\mathbb{UCQ}$: it is NP-complete, even for schemas of bounded arity [17]. This holds since CQS evaluation in $(\mathbb{C}, \mathbb{UCQ})$ is, in fact, a refinement of the query evaluation problem for $\mathbb{UCQ}$, with the additional promise that the input database satisfies a given (but potentially empty) set of integrity constrains coming from $\mathbb{C}$.

The Limits of Efficiency. Thus both OMQ evaluation and CQS evaluation in $(\mathbb{C}, \mathbb{UCQ})$, where $\mathbb{C} \in \{\mathbb{G}, \mathbb{FG}\}$, are computationally hard problems. However, there are subclasses of $(\mathbb{C}, \mathbb{UCQ})$, seen either as an OMQ language or a class of CQSs, for which the evaluation problem is efficient in the sense of being tractable or being fixed-parameter tractable (FPT) where the parameter is the size of the OMQ (resp., CQS). In particular, with $\mathbb{UCQ}_k$ being the class of UCQs of treewidth at most k, we know that: (1) for each $k \geq 1$, OMQ evaluation in $(\mathbb{G}, \mathbb{UCQ}_k)$ is in FPT (actually, this is shown in this work – Proposition 3.3), and (2) for each $k \geq 1$, CQS evaluation in $(\mathbb{FG}, \mathbb{UCQ}_k)$ is in PTime. Both statements rely on the well-known result that query evaluation for UCQs of bounded treewidth is tractable [18]. In view of this, it is natural to ask whether we can precisely characterize the limits of efficient OMQ and CQS evaluation. The goal of this work is to provide such efficiency characterizations.

A seminal result by Grohe precisely characterizes the (recursively enumerable) classes of CQs over schemas of bounded arity that can be evaluated in polynomial time (under the assumption that FPT $\neq$ W[1]) [26]: this is the case if and only if for some $k \geq 1$, every CQ in the class is equivalent to a CQ of treewidth k. Grohe's result also establishes that PTime and FPT coincide for evaluating classes of CQs. The above can be naturally generalized to UCQs.

Efficiency characterizations in the same spirit have been recently obtained for OMQ languages based on description logics (DLs). In particular, [7] precisely characterizes the (recursively enumerable) classes of OMQs from $(\mathcal{ELHI}_\bot, \mathbb{UCQ})$, where $\mathcal{ELHI}_\bot$ is an important DL, essentially a fragment of guarded TGDs. Here, OMQ evaluation is in FPT (assuming FPT $\neq$ W[1]) if and only if for some $k \geq 1$, every OMQ in the class is equivalent to an OMQ of treewidth k, i.e., an OMQ such that the UCQ in it is of treewidth k. Note that the equivalence is now on the level of OMQs rather than on the level of UCQs. The same work [7] precisely characterizes the (recursively enumerable) classes of OMQs from $(\mathcal{ELH}_\bot, \mathbb{UCQ})$, where $\mathcal{ELH}_\bot$ is a key fragment of $\mathcal{ELHI}_\bot$ that underpins the OWL 2 EL profile of the OWL 2 recommendation [32]. Here, evaluation is in PTime (again assuming FPT $\neq$ W[1]) if and only if for some $k \geq 1$, every OMQ in the class is equivalent to an OMQ of treewidth k. This also shows that PTime and FPT coincide for evaluating classes of OMQs from $(\mathcal{ELH}_\bot, \mathbb{UCQ})$. The classification of $(\mathcal{ELH}_\bot, \mathbb{UCQ})$, however, is subject to the condition that the ontology does not introduce relations beyond those admitted in the database.

Our Results. Our main results are as follows, under the widely believed complexity-theoretic assumption that FPT $\neq$ W[1]:

(1) For a recursively enumerable class of OMQs from $(\mathbb{G}, \mathbb{UCQ})$ over a schema of bounded arity, evaluation is in FPT if and only if there is $k \geq 1$ such that each OMQ in the class is equivalent to one from $(\mathbb{G}, \mathbb{UCQ}_k)$ (Theorem 5.3).

(2) For a recursively enumerable class of CQSs from $(\mathbb{G}, \mathbb{UCQ})$ over a schema of bounded arity, evaluation is in PTime if and only if evaluation is in FPT if and only if there is $k \geq 1$ such that each CQS in the class is equivalent to one from $(\mathbb{G}, \mathbb{UCQ}_k)$ (Theorem 5.7).

(3) Our final result concerns CQSs from $(\mathbb{FG}, \mathbb{UCQ})$ in which the TGDs have a bounded number of at most $m \geq 1$ head atoms; let $\mathbb{FG}_m$ be the obtained class. For a recursively enumerable class of CQSs from $(\mathbb{FG}_m, \mathbb{UCQ})$ over a schema of bounded arity, evaluation is in PTime if and only if evaluation is in FPT if and only if there is $k \geq 1$ such that each CQS in the class is equivalent to one from $(\mathbb{FG}_m, \mathbb{UCQ}_k)$ (Theorem 5.12).

It is not surprising that characterization (1) talks only about FPT and not PTime complexity since we know from [14] that the ExpTime-hardness of OMQ evaluation in $(\mathbb{G}, \mathbb{UCQ})$ in the case of schemas of bounded arity holds even if the actual query is an atomic query of the simplest form, i.e., a propositional atom. Moreover, it turns out that the notion of being equivalent to an OMQ of bounded treewdith is not enough for characterizing evaluation in FPT for frontier-guarded TGDs. We can show that OMQ evaluation in $(\mathbb{FG}, \mathbb{UCQ}_k)$ is W[1]-hard – this is actually easily inherited from the fact that Boolean CQ evaluation is W[1]-hard [33].

For all the above characterizations, if the efficiency condition fails, i.e., there is no integer $k \geq 1$ such that each OMQ or CQS in the considered class is equivalent to one in which the actually query falls within $\mathbb{UCQ}_k$, then the evaluation problem is W[1]-hard. Showing these lower bounds is actually the most challenging task underlying our efficiency characterizations. Together with the assumption that FPT $\neq$ W[1], they establish that efficient evaluation implies the efficiency condition. For (1) and (3), this is done via an fpt-reduction from the parameterized version of the k-clique problem, a well-known W[1]-hard problem, building on Grohe's result which also relies on an fpt-reduction from k-clique. For (2), we exploit a novel connection between OMQ evaluation and CQS evaluation, which is of independent interest. In fact, we provide an fpt-reduction from OMQ evaluation to CQS evaluation.

At this point, we would like to stress that the fpt-reductions underlying (1) and (3) are not merely adaptations of the fpt-reduction

by Grohe. For (1), the fact that the ontology can introduce additional relations beyond the data signature causes serious challenges. Moreover, unlike the characterizations of [7] for OMQs based on DLs, where only unary and binary relations are used, we need to deal with relations of arity beyond two, which also causes additional non-trivial complications that require novel ideas and techniques. For (3), unlike Grohe, we are more constrained in defining the right database as it must satisfy the given set of constraints. Moreover, the useful notion of core of a CQ, which was crucial for Grohe's proof, cannot be directly used in the presence of constraints.

We further study the complexity of the associated meta problems of deciding whether, for some fixed k, a given OMQ or CQS is equivalent to one in which the actual query falls within $\mathbb{UCQ}_k$. We show that in all the considered cases the problem is 2ExpTime-complete (under the mild assumption that k is at least the maximum arity of the occurring relation symbols, minus one). Note that decidability of the meta problem is needed to prove the lower bounds described above, but we also consider it interesting in its own right.

2 PRELIMINARIES

We consider the disjoint countably infinite sets $\mathbf{C}$ and $\mathbf{V}$ of *constants* and *variables*, respectively. We refer to constants and variables as *terms*. For an integer $n \geq 1$, we may write $[n]$ for the set $\{1, \ldots, n\}$.

Relational Databases. A *(relational) schema* $\mathbf{S}$ is a finite set of relation symbols (or predicates) with associated arity. We write $\mathsf{ar}(R)$ for the arity of a predicate R, and $\mathsf{ar}(\mathbf{S})$ for the arity of $\mathbf{S}$, that is, the number $\max_{R \in \mathbf{S}}\{\mathsf{ar}(R)\}$. An *atom* over $\mathbf{S}$ is an expression of the form $R(\bar{t})$, where $R \in \mathbf{S}$ and $\bar{t}$ is an $\mathsf{ar}(R)$-tuple of terms. An *instance* over $\mathbf{S}$, or simply $\mathbf{S}$-*instance*, is a (possibly infinite) set of atoms over $\mathbf{S}$ that contain only constants, while a *database* over $\mathbf{S}$, or simply $\mathbf{S}$-*database*, is a finite $\mathbf{S}$-instance. We write $\mathsf{dom}(I)$ for the set of constants in an instance I. For a set $T \subseteq \mathsf{dom}(I)$, we denote by $I_{|T}$ the restriction of I to atoms that mention only constants of T. A *homomorphism* from I to an instance J is a function $h : \mathsf{dom}(I) \to \mathsf{dom}(J)$ such that $R(h(\bar{t})) \in J$ for every $R(\bar{t}) \in I$. We write $I \to J$ for the fact that there is a homomorphism from I to J.

Conjunctive Queries. A *conjunctive query* (CQ) over a schema $\mathbf{S}$ is a first-order formula of the form $q(\bar{x}) := \exists \bar{y} \left(R_1(\bar{x}_1) \wedge \cdots \wedge R_m(\bar{x}_m)\right)$, where each $R_i(\bar{x}_i)$, for $i \in [m]$, is an atom over $\mathbf{S}$ that contains only variables, each variable mentioned in the $\bar{x}_i$s appears either in $\bar{x}$ or $\bar{y}$, and $\bar{x}$ contains all the free variables of q called the *answer variables*. Every CQ q can be naturally seen as a database $D[q]$, known as the *canonical database* of q, obtained by dropping the existential quantifier prefix and viewing variables as constants. We may simply write q instead of $D[q]$. A homomorphism from a CQ q to an instance I is a homomorphism from $D[q]$ to I. A tuple $\bar{c} \in \mathsf{dom}(I)^{|\bar{x}|}$ is an *answer* to Q over I if there is a homomorphism h from q to I with $h(\bar{x}) = \bar{c}$. The *evaluation of* $q(\bar{x})$ *over* I, denoted $q(I)$, is the set of all answers to Q over I. We write $\mathbb{CQ}$ for the class of CQs. A *union of conjunctive queries* (UCQ) over a schema $\mathbf{S}$ is a first-order formula of the form $q(\bar{x}) := q_1(\bar{x}) \vee \cdots \vee q_n(\bar{x})$, where $n \geq 1$, and $q_i(\bar{x})$, for $i \in [n]$, is a CQ over $\mathbf{S}$. The evaluation of q over an instance I, denoted $q(I)$, is defined as the set of tuples $\bigcup_{i \in [n]} q_i(I)$. We write $\mathbb{UCQ}$ for the class of UCQs. The *arity* of a (U)CQ is defined as the number of its answer variables. A (U)CQ of arity zero is called *Boolean*, and it can have as an answer only the empty tuple. For a Boolean (U)CQ q, we may write $I \models q$, if $q(I) = \{()\}$, and $I \not\models q$, otherwise.

Treewidth. A central notion in our work is that of treewidth, which measures the degree of tree-likeness of a graph. Let $G = (V, E)$ be an undirected graph. A *tree decomposition* of G is a pair $\delta = (T_\delta, \chi)$, where $T_\delta = (V_\delta, E_\delta)$ is a tree, and χ is a labeling function $V_\delta \to 2^V$, i.e., χ assigns a subset of V to each node of T_δ, such that:

(1) $\bigcup_{t \in V_\delta} \chi(t) = V$.
(2) If $\{u, v\} \in E$, then $u, v \in \chi(t)$ for some $t \in V_\delta$.
(3) For each $v \in V$, the set of nodes $\{t \in V_\delta \mid v \in \chi(t)\}$ induces a connected subtree of T_δ.

The *width* of δ is the number $\max_{t \in V_\delta}\{|\chi(t)|\} - 1$. If the edge-set E of G is non-empty, then the *treewidth* of G is the minimum width over all its tree decompositions; otherwise, it is defined to be one. Each instance I is associated with an undirected graph (without self loops) $G^I = (V, E)$, called the *Gaifman graph* of I, defined as follows: $V = \mathsf{dom}(I)$, and $\{a, b\} \in E$ iff there is an atom $R(\bar{t}) \in I$ that mentions both a and b. The treewidth of I is the treewidth of G^I. For $k \geq 1$, TW_k is the class of instances of treewidth at most k.

The *evaluation problem* for (U)CQs takes as input a (U)CQ $q(\bar{x})$, a database D, and a candidate answer $\bar{c}$, and asks whether $\bar{c} \in q(D)$. It is well-known that (U)CQ evaluation is NP-complete [17]. On the other hand, it becomes tractable by restricting the syntactic shape of CQs. One of the most widely studied such restrictions is bounded treewidth. Formally, a CQ $q(\bar{x}) = \exists \bar{y}\, \phi(\bar{x}, \bar{y})$ has treewidth $k \geq 1$ if $G^q_{|\bar{y}}$ has treewidth k, where $G^q_{|\bar{y}}$ is the subgraph of G^q induced by the elements of $\bar{y}$. Note that the treewidth of q is defined in a more liberal way than usual. The standard definition considers the treewitdh of G^q, while here the treewidth of q is only measured with respect to the subgraph of G^q induced by its existentially quantified variables. A UCQ q has treewidth k if each of its disjuncts has treewidth at most k. We write $\mathbb{CQ}_k$ (resp., $\mathbb{UCQ}_k$) for the class of CQs (resp. UCQs) of treewidth at most $k \geq 1$. The next well-known result illustrates the usefulness of bounding the treewidth.

Proposition 2.1 ([18]). *Fix $k \geq 1$. Given a database D, an n-ary query $q \in \mathbb{CQ}_k$, and a tuple $\bar{c} \in \mathsf{dom}(D)^n$, the problem of deciding whether $\bar{c} \in q(D)$ can be solved in time $O(||D||^{k+1} \cdot ||q||)$.*[1]

Tuple-generating Dependencies. A *tuple-generating dependency* (TGD) σ over $\mathbf{S}$ is a constant-free first-order sentence of the form $\forall \bar{x} \forall \bar{y} \left(\phi(\bar{x}, \bar{y}) \rightarrow \exists \bar{z}\, \psi(\bar{x}, \bar{z})\right)$, where ϕ is a possibly empty conjunction of atoms over $\mathbf{S}$, while ψ is a non-empty conjunction of atoms over $\mathbf{S}$. For simplicity, we write σ as $\phi(\bar{x}, \bar{y}) \rightarrow \exists \bar{z}\, \psi(\bar{x}, \bar{z})$, and use comma instead of $\wedge$ for joining atoms. We call ϕ and ψ the *body* and *head* of σ, denoted $\mathsf{body}(\sigma)$ and $\mathsf{head}(\sigma)$, respectively. The *frontier* of σ, denoted $\mathsf{fr}(\sigma)$, is the set of variables $\bar{x}$, i.e., the variables that appear both in the body and the head of σ. The TGD σ above is logically equivalent to the expression $\forall \bar{x}\, (q_\phi(\bar{x}) \rightarrow q_\psi(\bar{x}))$, where $q_\phi(\bar{x})$ and $q_\psi(\bar{x})$ are the CQs $\exists \bar{y}\, \phi(\bar{x}, \bar{y})$ and $\exists \bar{z}\, \psi(\bar{x}, \bar{z})$, respectively. Therefore, an instance I over $\mathbf{S}$ satisfies σ, denoted $I \models \sigma$, if $q_\phi(I) \subseteq q_\psi(I)$. An instance I *satisfies* a set Σ of TGDs, denoted $I \models \Sigma$, if $I \models \sigma$ for each $\sigma \in \Sigma$. Henceforth, whenever we refer to a set of TGDs we mean a finite set. We write $\mathbb{TGD}$ for the class of TGDs, that is, the family of all possible sets of TGDs.

[1] As usual, given a syntactic object O, we write $||O||$ for its size.

Frontier-Guardedness. A TGD σ is *guarded* if either body(σ) is empty, or there exists an atom α in its body that contains all the variables occurring in body(σ) [14]. Such an atom α is the *guard* of σ, denoted guard(σ). We write $\mathbb{G}$ for the class of guarded TGDs. A natural generalization of guardedness is frontier-guardedness, where only the frontier variables must be guarded. Formally, a TGD σ is *frontier-guarded* if either body(σ) is empty or there exists an atom α in its body that contains all the variables of fr(σ) [3], which we call again guard and denote as guard(σ). The class of frontier-guarded TGDs is denoted by $\mathbb{FG}$. Clearly, $\mathbb{G} \subsetneq \mathbb{FG} \subsetneq \mathbb{TGD}$.

The Chase Procedure. The *chase* is a useful tool when reasoning with TGDs [14, 22, 28, 30]. We first define a single chase step. Let I be an instance over a schema $\mathbf{S}$ and σ a TGD of the form $\phi(\bar{x}, \bar{y}) \rightarrow \exists \bar{z}\, \psi(\bar{x}, \bar{z})$ over $\mathbf{S}$. We say that σ is *applicable* with respect to I, if there exists a tuple $(\bar{c}, \bar{c}')$ of constants in I such that $\phi(\bar{c}, \bar{c}') \subseteq I$. In this case, *the result of applying σ over I with* $(\bar{c}, \bar{c}')$ is the instance $J = I \cup \psi(\bar{c}, \bar{c}'')$, where $\bar{c}''$ is the tuple obtained from $\bar{z}$ by simultaneously replacing each variable z with a fresh distinct constant not occurring in I. For such a single chase step we write $I \xrightarrow{\sigma, (\bar{c}, \bar{c}')} J$.

Let I be an instance and Σ a set of TGDs. A *chase sequence for I under Σ* is a sequence of chase steps $I_0 \xrightarrow{\sigma_0, (\bar{c}_0, \bar{c}'_0)} I_1 \xrightarrow{\sigma_1, (\bar{c}_1, \bar{c}'_1)} I_2 \ldots$ such that (1) $I_0 = I$, (2) $\sigma_i \in \Sigma$ for each $i \geq 0$, and (3) $J \models \Sigma$, where $J = \bigcup_{i \geq 0} I_i$. The instance J is the (potentially infinite) *result* of this chase sequence, which always exists. Since we consider the oblivious chase, i.e., a TGD is triggered whenever its body is satisfied no matter whether its head is satisfied, every chase sequence for I under Σ leads to the same result (up to isomorphism). Thus, we can refer to *the* result of the chase for I under Σ, denoted chase(I, Σ). The key property of the chase follows:

Proposition 2.2. *Consider an instance I and a set Σ of TGDs. For every instance J such that $J \supseteq I$ and $J \models \Sigma$,* chase(I, Σ) $\rightarrow J$ *via a homomorphism that is the identity on* dom(I).

Parameterized Complexity. Parameterized complexity has a central role in our work. A *parameterized problem* over an alphabet Λ is a pair (P, κ), with $P \subseteq \Lambda^*$ a decision problem and κ a *parameterization* of P, that is, a PTime computable function $\kappa : \Lambda^* \rightarrow \mathbb{N}$. A prime example is p-Clique, where P is the set of all pairs (G, k) with G an undirected graph that contains a k-clique and $\kappa(G, k) = k$.

A problem (P, κ) is *fixed-parameter tractable* (fpt) if there is a computable function $f : \mathbb{N} \rightarrow \mathbb{N}$ and an algorithm that decides P in time $|x|^{O(1)} \cdot f(\kappa(x))$, where x denotes the input. We use FPT to denote the class of all parameterized problems that are fixed-parameter tractable. Notice that FPT corresponds to a relaxation of the usual notion of tractability: if P is in PTime, then (P, κ) is in FPT for any κ, but the latter might also be the case when P is NP-hard.

An *fpt-reduction* from a problem (P_1, κ_1) over Λ_1 to a problem (P_2, κ_2) over Λ_2 is a function $\rho : \Lambda_1^* \rightarrow \Lambda_2^*$ such that, for some computable functions $f, g : \mathbb{N} \rightarrow \mathbb{N}$,

(1) $x \in P_1$ iff $\rho(x) \in P_2$, for all $x \in \Sigma_1^*$;
(2) $\rho(x)$ is computable in time $||x||^{O(1)} \cdot f(\kappa_1(x))$, for $x \in \Lambda_1^*$;
(3) $\kappa_2(\rho(x)) \leq g(\kappa_1(x))$, for all $x \in \Lambda_1^*$.

An important parameterized complexity class is W[1] $\supseteq$ FPT. Hardness for W[1] is defined in terms of fpt-reductions. It is generally believed that FPT $\neq$ W[1], the status of this problem being comparable to that of PTime $\neq$ NP. Hence, if a parameterized problem (P, κ) is W[1]-hard, then (P, κ) is not fpt unless FPT = W[1]. A well-known W[1]-hard problem is p-Clique [21].

3 THE TWO FACETS OF TGDS IN QUERYING

As discussed in Section 1, TGDs can be used as:

(1) *Ontology axioms* that enrich incomplete data with domain knowledge, which leads to more complete answers.
(2) *Integrity constraints* that specify semantic properties satisfied by all databases, which paves the way to constraint-aware query optimization techniques.

Depending on how TGDs are used, we obtain different evaluation problems, which we now formalize.

3.1 TGDs as Ontology Axioms

When a set of TGDs is used as ontology axioms, or an ontology, it is typically seen, together with the actual query, as one composite query, called ontology-mediated query. Formally, an *ontology-mediated query* (OMQ) is a triple $Q = (\mathbf{S}, \Sigma, q)$, where $\mathbf{S}$ is a schema, called *data schema*, which indicates that Q will be evaluated over $\mathbf{S}$-databases, Σ is a set of TGDs over an extended schema $\mathbf{T} \supseteq \mathbf{S}$, called *ontology*, and q is a UCQ over $\mathbf{T}$. In case $\mathbf{S} = \mathbf{T}$, we say that Q has *full data schema*. The *arity* of Q is defined as the arity of q. We write $Q(\bar{x})$ to emphasize that the answer variables of q are $\bar{x}$, and say that Q is *over* $\mathbf{T}$ to mean that the extended schema of Q is $\mathbf{T}$.

The evaluation of an OMQ $Q(\bar{x}) = (\mathbf{S}, \Sigma, q(\bar{x}))$ over an $\mathbf{S}$-database D consists of all the tuples that are answers to q over each model of D and Σ. A *model* of D and Σ is an instance I such that $I \supseteq D$ and $I \models \Sigma$. A tuple $\bar{c} \in \text{dom}(D)^{|\bar{x}|}$ is an *answer* to Q over D if $\bar{c} \in q(I)$ for each model I of D and Σ. The *evaluation of* $Q(\bar{x})$ over D, denoted $Q(D)$, is the set of all answers to Q over D. By Proposition 2.2, and the monotonicity of UCQs, we get the following useful result:

Proposition 3.1. *For every OMQ $Q = (\mathbf{S}, \Sigma, q) \in (\mathbb{TGD}, \mathbb{UCQ})$, $Q(D) = q(\text{chase}(D, \Sigma))$.*

We write $(\mathbb{C}, \mathbb{Q})$ for the class of OMQs, called *OMQ language*, in which the ontology is formulated in the class of TGDs $\mathbb{C}$, and the actual query is coming from the class of queries $\mathbb{Q}$; for example, we may write $(\mathbb{G}, \mathbb{CQ})$, $(\mathbb{FG}, \mathbb{UCQ})$, $(\mathbb{G}, \mathbb{UCQ}_k)$, for some $k \geq 1$, etc. This brings us to the evaluation problem for OMQ languages $\mathbb{O}$:

PROBLEM :	OMQ-Evaluation($\mathbb{O}$)
INPUT :	An OMQ $Q = (\mathbf{S}, \Sigma, q(\bar{x})) \in \mathbb{O}$, an $\mathbf{S}$-database D, and a tuple $\bar{c} \in \text{dom}(D)^{\lvert\bar{x}\rvert}$
QUESTION :	Is it the case that $\bar{c} \in Q(D)$?

We are also interested in the parameterized version of the above problem, dubbed p-OMQ-Evaluation($\mathbb{O}$), with the parameter being the size of the OMQ Q, as customary in the literature [33]. Thus p-OMQ-Evaluation($\mathbb{O}$) is in FPT if it can be solved in time $||D||^{O(1)} \cdot f(||Q||)$ for some computable function $f : \mathbb{N} \rightarrow \mathbb{N}$.

It is well-known that OMQ-Evaluation($\mathbb{TGD}, \mathbb{CQ}$) is undecidable; see, e.g., [14]. On the other hand, if we focus on OMQs in which the ontology is formulated as a set of frontier-guarded TGDs, then the problem becomes decidable, in fact, it is 2ExpTime-complete [2]. At this point, one may wonder whether bounding the treewidth

of the CQs will have the same positive effect as in the case of CQ evaluation (see Proposition 2.1). It is implicit in [14] that this is not the case. The next result summarizes some key facts about the evaluation problem for OMQs based on (frontier-)guarded TGDs.

Proposition 3.2. *It holds that:*

(1) OMQ-Evaluation($\mathbb{FG}, \mathbb{UCQ}$) *is* 2ExpTime*-complete even for schemas of bounded arity.*
(2) OMQ-Evaluation($\mathbb{G}, \mathbb{UCQ}$) *is* 2ExpTime*-complete, and becomes* ExpTime*-complete for schemas of bounded arity.*
(3) For each $k \geq 1$, OMQ-Evaluation($\mathbb{G}, \mathbb{CQ}_k$) *is* 2ExpTime*-complete, and still* ExpTime*-hard for schemas of bounded arity.*

Since the parameterized version of the evaluation problem for CQs is W[1]-hard [33], even for schemas of bounded arity, we can immediately conclude that p-OMQ-Evaluation($\mathbb{G}, \mathbb{CQ}$) is W[1]-hard, even for schemas of bounded arity. Do we gain something if we focus on UCQs of bounded treewidth? The answer is negative for frontier-guarded TGDs, but affirmative for guarded TGDs.

Proposition 3.3. *It holds that:*

(1) p-OMQ-Evaluation($\mathbb{G}, \mathbb{CQ}$) *is* W[1]*-hard even for schemas of bounded arity.*
(2) p-OMQ-Evaluation($\mathbb{FG}, \mathbb{CQ}_k$) *is* W[1]*-hard even for schemas of bounded arity.*
(3) For each $k \geq 1$, p-OMQ-Evaluation($\mathbb{G}, \mathbb{UCQ}_k$) *is in* FPT.

As discussed above, item (1) is a consequence of the fact that the parameterized version of CQ evaluation is W[1]-hard, even for schemas of bounded arity. Item (2) is a consequence of the fact that the parameterized version of CQ evaluation is W[1]-hard, even for Boolean CQs over a schema of bounded arity, while a Boolean CQ can be transformed into a frontier-guarded TGD. Indeed, given a Boolean CQ $\exists \bar{x}\, \phi(\bar{x})$, $\phi(\bar{x}) \rightarrow \text{Ans}$, where Ans is a 0-ary predicate, is trivially a frontier-guarded TGD since its frontier is empty.

We now briefly explain how item (3) is shown, while the details can be found in the appendix. A TGD is called *linear* if it has only one atom in its body, while the class of linear TGDs is denoted $\mathbb{L}$. The key ingredient underlying item (3) is that, given an $\mathbf{S}$-database D and an OMQ $Q = (\mathbf{S}, \Sigma, q)$ from $(\mathbb{G}, \mathbb{UCQ})$, $Q(D)$ coincides with the evaluation of q over an initial finite portion C of $\mathsf{chase}(D^*, \Sigma^*)$, where D^* can be computed from D and Σ, and $\Sigma^* \in \mathbb{L}$ can be computed solely from Σ. Roughly, C is the finite instance obtained by keeping only the atoms of $\mathsf{chase}(D^*, \Sigma^*)$ up to a finite level that depends only on Σ and q, while the notion of level indicates the distance of an atom in the chase from the starting database. Furthermore, the instance C can be computed in time $||D||^{O(1)} \cdot f(||Q||)$ for some computable triple exponential function $f : \mathbb{N} \rightarrow \mathbb{N}$. Therefore, to decide whether a tuple $\bar{c}$ over $\mathsf{dom}(D)$ belongs to $Q(D)$, it suffices to construct the finite instance C, and accept if $\bar{c}$ belongs to $q(C)$; otherwise, reject. Since $q \in \mathbb{UCQ}_k$, by Proposition 2.1, the overall procedure takes time $||D||^{O(1)} \cdot g(||Q||)$ for some computable triple exponential function $g : \mathbb{N} \rightarrow \mathbb{N}$, and the claim follows.

3.2 TGDs as Integrity Constraints

When a set of TGDs is used as integrity constraints, the problem that we are interested in is simply a refinement of the standard query evaluation problem, with the additional promise that the input database satisfies the given set of TGDs. To this end, the evaluation problem is parameterized, not only with the query language in question, but also with the class of TGDs from which the constraints are coming. Formally, a *constraint-query specification* (CQS) over a schema $\mathbf{T}$ is a pair $S = (\Sigma, q)$, where Σ is a set of TGDs over $\mathbf{T}$, the set of *integrity constraints*, and q a UCQ over $\mathbf{T}$. We overload the notation and write $(\mathbb{C}, \mathbb{Q})$ for the class of CQSs in which the set of integrity constraints is formulated in the class of TGDs $\mathbb{C}$, and the query is coming from the class of queries $\mathbb{Q}$. It will be clear from the context whether $(\mathbb{C}, \mathbb{Q})$ is an OMQ language or a class of CQSs. The evaluation problem for CQSs follows:

PROBLEM :	CQS-Evaluation($\mathbb{O}$)
INPUT :	A CQS $S = (\Sigma, q(\bar{x})) \in \mathbb{O}$ over a schema $\mathbf{T}$, an $\mathbf{T}$-database D such that $D \models \Sigma$, and a tuple $\bar{c} \in \mathsf{dom}(D)^{\lvert\bar{x}\rvert}$
QUESTION :	Is it the case that $\bar{c} \in q(D)$?

We are also interested in the parameterized version of the above problem, which we call p-CQS-Evaluation($\mathbb{O}$), with the parameter being the size of the CQS S. Therefore, p-CQS-Evaluation($\mathbb{O}$) is in FPT if it can be solved in time $||D||^{O(1)} \cdot f(||S||)$ for some computable function $f : \mathbb{N} \rightarrow \mathbb{N}$.

Recall that the evaluation problem for CQs is NP-hard [17], while its parameterized version is W[1]-hard [33], even for schemas of bounded arity. Hence, the same holds for CQS-Evaluation($\mathbb{G}, \mathbb{CQ}$). Now, bounding the treewidth of the CQs has the same positive effect as in the case of CQ evaluation (see Proposition 2.1). In fact, CQS-Evaluation($\mathbb{FG}, \mathbb{UCQ}_k$) is in PTime. Thus, the evaluation problem for UCQs, and the evaluation problem for the classes of CQSs based on $\mathbb{G}$ and $\mathbb{FG}$ have the same complexity.

4 SEMANTIC TREE-LIKENESS

A seminal result by Grohe precisely characterizes the (recursively enumerable) classes of CQs over schemas of bounded arity that can be evaluated in polynomial time (under the assumption that FPT $\neq$ W[1]). In fact, this result shows that the classes of CQs over schemas of bounded arity that can be evaluated in polynomial time are precisely those that are semantically tree-like, or, more formally, are of *bounded treewidth modulo equivalence*. Moreover, the result states that fpt does not add anything to standard tractability in the considered setting. Before giving the statement, let us formalize the notion of bounded treewidth modulo equivalence. Recall that two CQs q, q' over a schema $\mathbf{S}$ are *equivalent* if, for every $\mathbf{S}$-database D, $q(D) = q'(D)$. For each $k \geq 1$, let $\mathbb{CQ}_k^{\equiv}$ be the class of all CQs that are equivalent to a CQ from $\mathbb{CQ}_k$. Grohe's result follows:

Theorem 4.1 (Grohe's Theorem [26]). *Fix* $r \geq 1$. *Let* $\mathbb{Q}$ *be a recursively enumerable class of CQs over schemas of arity* r. *The following are equivalent, assuming* FPT $\neq$ W[1]*:*

(1) The evaluation problem for $\mathbb{Q}$ *is in* PTime.
(2) The evaluation problem for $\mathbb{Q}$ *is in* FPT.
(3) There is $k \geq 1$ *such that* $\mathbb{Q} \subseteq \mathbb{CQ}_k^{\equiv}$.

If either statements is false, then evaluation for $\mathbb{Q}$ *is* W[1]*-hard.*

Interestingly, it is decidable whether a CQ is equivalent to one of treewidth k. This is shown by exploiting the notion of core.

Recall that the *core* of a CQ q is a $\subseteq$-minimal subquery of q that is equivalent to q. It is known that, for each $k \geq 1$, a CQ q belongs to $\mathbb{CQ}_k^{\equiv}$ iff its core is in $\mathbb{CQ}_k$, and that deciding this property is NP-complete [20]. There is also a natural generalization of this characterization and of Theorem 4.1 to the class of UCQs.

At this point, it is natural to ask whether it is possible to obtain a characterization of the classes of OMQs (resp., CQSs) based on (frontier-)guarded TGDs for which OMQ-Evaluation (resp., CQS-Evaluation) can be efficiently solved, in the same spirit as Grohe's Theorem. In fact, our main question is whether the natural generalization of the notion of bounded treewidth modulo equivalence for CQs to OMQs (resp., CQSs) is a decidable notion that exhausts tractability or fixed-parameter tractability for OMQ-Evaluation (resp., CQS-Evaluation), as in the case of Grohe's Theorem.

4.1 Semantic Tree-likeness for OMQs

We first concentrate on OMQs and introduce the notion of $\mathbb{UCQ}_k$-equivalence, which essentially tells us that an OMQ can be rewritten into an equivalent one where the UCQ belongs to $\mathbb{UCQ}_k$. Let us clarify that for OMQs the notion of equivalence is not applied at the level of the actual query, but to the whole OMQ. Formally, given two OMQs Q and Q', both with data schema $\mathbf{S}$, Q is *contained* in Q', written $Q \subseteq Q'$, if $Q(D) \subseteq Q'(D)$ for every $\mathbf{S}$-database D. We then say that Q and Q' are *equivalent*, denoted $Q \equiv Q'$, if $Q \subseteq Q'$ and $Q' \subseteq Q$. In what follows, let $\mathbb{C}$ be a class of TGDs, i.e., $\mathbb{C} \subseteq \mathbb{TGD}$.

Definition 4.2 (**$\mathbb{UCQ}_k$-equivalence for OMQs**). An OMQ $Q = (\mathbf{S}, \Sigma, q) \in (\mathbb{C}, \mathbb{UCQ})$ is *$\mathbb{UCQ}_k$-equivalent*, for $k \geq 1$, if there exists an OMQ $Q' = (\mathbf{S}, \Sigma', q') \in (\mathbb{C}, \mathbb{UCQ}_k)$ such that $Q \equiv Q'$. Given an OMQ language $\mathbb{O} = (\mathbb{C}, \mathbb{UCQ})$, for each $k \geq 1$, let $\mathbb{O}_k^{\equiv}$ be the class of all OMQs from $\mathbb{O}$ that are $\mathbb{UCQ}_k$-equivalent. ∎

According to the above definition, we are allowed to rewrite both the ontology and the UCQ. It is conceptually meaningful though to consider also the setting where the ontology cannot be altered. This leads to the uniform version of $\mathbb{UCQ}_k$-equivalence.

Definition 4.3 (**Uniform $\mathbb{UCQ}_k$-equivalence for OMQs**). An OMQ $Q = (\mathbf{S}, \Sigma, q) \in (\mathbb{C}, \mathbb{UCQ})$ is *uniformly $\mathbb{UCQ}_k$-equivalent*, for $k \geq 1$, if there is $Q' = (\mathbf{S}, \Sigma, q') \in (\mathbb{C}, \mathbb{UCQ}_k)$ such that $Q \equiv Q'$. For an OMQ language $\mathbb{O} = (\mathbb{C}, \mathbb{UCQ})$ and $k \geq 1$, let $\mathbb{O}_k^{\equiv,u}$ be the class of all OMQs from $\mathbb{O}$ that are uniformly $\mathbb{UCQ}_k$-equivalent. ∎

The next example shows that both the ontology and the data schema can have an impact on the treewidth.

Example 4.4. We first illustrate that the ontology can have an impact on the treewidth. Consider the OMQ $Q_1 = (\mathbf{S}, \Sigma, q)$, where

$$\begin{aligned} \mathbf{S} &= \{R_1, R_2, R_3, R_4, P\} \\ \Sigma &= \{R_2(x) \rightarrow R_4(x)\} \\ q() &= P(x_2, x_1) \wedge P(x_4, x_1) \wedge P(x_2, x_3) \wedge P(x_4, x_3) \wedge \\ &\quad R_1(x_1) \wedge R_2(x_2) \wedge R_3(x_3) \wedge R_4(x_4). \end{aligned}$$

Note that $\mathbf{S}$ contains all the predicates in Σ and q, while q is a Boolean CQ (the existential quantifiers are omitted). Observe that q is a core from $\mathbb{CQ}_2$, and thus can easily be seen to not belong to $\mathbb{UCQ}_1^{\equiv}$. However, Q_1 is equivalent to the OMQ $(\mathbf{S}, \Sigma, q')$, where

$$q'() = P(x_2, x_1) \wedge P(x_2, x_3) \wedge R_1(x_1) \wedge R_2(x_2) \wedge R_3(x_3).$$

Since $q' \in \mathbb{CQ}_1$, we get that $Q_1 \in (\mathbb{G}, \mathbb{UCQ})_1^{\equiv,u}$.

We proceed to show that the data schema can also have an impact. Consider the OMQ $Q_2 = (\mathbf{S}', \Sigma', q)$ with full data schema, where

$$\Sigma' = \{S(x) \rightarrow R_1(x), \quad S(x) \rightarrow R_3(x)\}.$$

It is not hard to see that Q_2 does not belong to $(\mathbb{G}, \mathbb{UCQ})_1^{\equiv}$. If, however, the predicate R_1 is omitted from the signature, then Q_2 is equivalent to the OMQ $(\{S, P, R_2, R_3, R_4\}, \Sigma', q'')$, where

$$q''() = P(x_2, x_1) \wedge P(x_4, x_1) \wedge R_1(x_1) \wedge R_2(x_2) \wedge R_3(x_1) \wedge R_4(x_4),$$

and thus, it belongs to $(\mathbb{G}, \mathbb{UCQ})_1^{\equiv,u}$.

4.2 Semantic Tree-likeness for CQSs

We now introduce $\mathbb{UCQ}_k$-equivalence for CQSs. For this setting, only the uniform version is relevant. Indeed, given a CQS $S = (\Sigma, q)$, by altering the set of integrity constraints Σ, we essentially change the semantics of S which is, of course, not our intention. Given two CQSs $S = (\Sigma, q)$ and $S' = (\Sigma, q')$ over a schema $\mathbf{T}$, we say that S is *contained* in S', denoted $S \subseteq S'$, if $q(D) \subseteq q'(D)$ for every $\mathbf{T}$-database D that satisfies Σ. We then say that S is *equivalent* to S' if $S \subseteq S'$ and $S' \subseteq S$. We may also write $q \subseteq_\Sigma q'$ (resp., $q \equiv_\Sigma q'$) for the fact that $S \subseteq S'$ (resp., $S \equiv S'$). We recall a known result about containment among CQSs, which will be useful for our analysis:

Proposition 4.5. *Let $S_1 = (\Sigma, q_1(\bar{x}))$ and $S_2 = (\Sigma, q_2(\bar{y}))$ be CQSs from $(\mathbb{TGD}, \mathbb{UCQ})$ with $|\bar{x}| = |\bar{y}|$. Then $S_1 \subseteq S_2$ iff for each $p_1 \in q_1$, there exists $p_2 \in q_2$ such that $\bar{x} \in p_2(\mathsf{chase}(p_1, \Sigma))$.*

The notion of uniform $\mathbb{UCQ}_k$-equivalence for CQSs follows:

Definition 4.6 (**Uniform $\mathbb{UCQ}_k$-equivalence for CQSs**). A CQS $S = (\Sigma, q)$ from $(\mathbb{C}, \mathbb{UCQ})$ is *uniformly $\mathbb{UCQ}_k$-equivalent*, for $k \geq 1$, if there exists a CQS $S' = (\Sigma, q')$ from $(\mathbb{C}, \mathbb{UCQ}_k)$ such that $S \equiv S'$. Given a class of CQSs $\mathbb{O}$, for $k \geq 1$, let $\mathbb{O}_k^{\equiv}$ be the class of all CQSs from $\mathbb{O}$ that are uniformly $\mathbb{UCQ}_k$-equivalent.[2] ∎

Observe that the first part of Example 4.4 also works when (Σ, q) is viewed as a CQS. This illustrates the fact that integrity constraints can have an impact on the treewidth.

5 OUR RESULTS IN A NUTSHELL

Having the relevant notions in place, we can now provide an answer to our main question, that is, whether (uniform) $\mathbb{UCQ}_k$-equivalence of OMQs (resp., uniform $\mathbb{UCQ}_k$-equivalence of CQSs) exhausts tractability and fixed-parameter tractability of OMQ-Evaluation (resp., CQS-Evaluation) in the same spirit as Grohe's Theorem. In this section, we give an overview of our results, observe interesting connections between the OMQ and CQS settings, and study the associated meta problems. Further details are deferred to the subsequent sections and the appendix.

5.1 The Guarded Case

Ontology-mediated Queries. We start with our results on OMQs based on guarded TGDs. We first ask whether (uniform) $\mathbb{UCQ}_k$-equivalence is decidable for $(\mathbb{G}, \mathbb{UCQ})$.

[2] We overload the notation again. It would be clear from the context when $\mathbb{O}_k^{\equiv}$ is an OMQ language or a class of CQSs. We also avoid the superindex u since for CQSs we only consider the uniform version of $\mathbb{UCQ}_k$-equivalence.

Theorem 5.1. *For each k, deciding whether a given OMQ Q from $(\mathbb{G}, \mathbb{UCQ})$ over $\mathbf{T}$ with $k \geq \mathsf{ar}(\mathbf{T}) - 1$ is (uniformly) $\mathbb{UCQ}_k$-equivalent is* 2ExpTime*-complete. If existant, an OMQ $Q' \in (\mathbb{G}, \mathbb{UCQ}_k)$ such that $Q \equiv Q'$ can be computed in double exponential time.*

The above complexity result exploits a characterization of when an OMQ Q from $(\mathbb{G}, \mathbb{UCQ})$ is (uniformly) $\mathbb{UCQ}_k$-equivalent, which in turn relies on what we call a $\mathbb{UCQ}_k$-approximation, that is, approximations of Q from below in terms of an OMQ from $(\mathbb{G}, \mathbb{UCQ}_k)$. In a nutshell, a $\mathbb{UCQ}_k$-approximation of $Q = (\mathbf{S}, \Sigma, q)$ is an OMQ $Q_k^a = (\mathbf{S}, \Sigma, q_k^a)$, where q_k^a belongs to $\mathbb{UCQ}_k$, that behaves like Q over $\mathbf{S}$-databases of treewidth at most k, i.e., for every $\mathbf{S}$-database D of treewidth at most k, $Q(D) = Q_k^a(D)$. It follows that Q_k^a is equivalent to Q if and only if Q is $\mathbb{UCQ}_k$-equivalent. The formal definition of Q_k^a can be found in the appendix. It is signficantly more involved than in the case of description logics [7] because there the chase only generates structures of treewidth one.

Proposition 5.2. *Let Q be an OMQ from $(\mathbb{G}, \mathbb{UCQ})$ over $\mathbf{T}$, and let $k \geq \mathsf{ar}(\mathbf{T}) - 1$. The following are equivalent:*

(1) Q is $\mathbb{UCQ}_k$-equivalent.
(2) Q is uniformly $\mathbb{UCQ}_k$-equivalent.
(3) $Q \equiv Q_k^a$.

Since $\mathbb{UCQ}_k$-equivalence and uniform $\mathbb{UCQ}_k$-equivalence turn out to be equivalent, we henceforth only use $\mathbb{UCQ}_k$-equivalence. Note that Proposition 5.2 also provides an approach to deciding $\mathbb{UCQ}_k$-equivalence, and thus to establishing Theorem 5.1: compute the $\mathbb{UCQ}_k$-approximation Q_k^a of Q and accept if $Q \subseteq Q_k^a$ (note that $Q_k^a \subseteq Q$ holds always); otherwise, reject. We can show that Q_k^a can be computed in double exponential time and it is known that OMQ containment for $(\mathbb{G}, \mathbb{UCQ})$ can be decided in double exponential time [6]. A naive use of these observations yields only a 4ExpTime upper bound, but we show in the appendix how to improve this to 2ExpTime. The lower bound is inherited from [7], where the same problem for OMQs based on DLs has been studied.

We remark that the case $k < \mathsf{ar}(\mathbf{T}) - 1$, excluded in Theorem 5.1 and Proposition 5.2, is somewhat esoteric as there are relation symbols whose arity is so high that they cannot be part of the UCQ in a $\mathbb{UCQ}_k$-approximation (unless variables are reused). In the appendix, we provide a concrete example which illustrates that this case is significantly different from the case $k \geq \mathsf{ar}(\mathbf{T}) - 1$; in particular, we can see that Proposition 5.2 is provably wrong.

We now state our main result concerning guarded OMQs, which shows that $\mathbb{UCQ}_k$-equivalence characterizes fpt for classes of OMQs from $(\mathbb{G}, \mathbb{UCQ})$ over schemas of bounded arity.

Theorem 5.3 (**Main Result I**). *Fix $r \geq 1$. Let $\mathbb{O}$ be a recursively enumerable class of OMQs from $(\mathbb{G}, \mathbb{UCQ})$ over a schema of arity r. The following are equivalent, assuming* FPT $\neq$ W[1]*:*

(1) p-OMQ-Evaluation($\mathbb{O}$) *is in* FPT.
(2) There is $k \geq 1$ such that $\mathbb{O} \subseteq (\mathbb{G}, \mathbb{UCQ})_k^{\equiv}$.

If either statement is false, then p-OMQ-Evaluation($\mathbb{O}$) *is* W[1]*-hard.*

The easy direction is (2) implies (1), which exploits Theorem 5.1, and the third item of Proposition 3.3. The hard task is to show that (1) implies (2). To this end, since we assume that FPT $\neq$ W[1], it suffices to show the following lower bound, which is our main technical result on OMQs based on guarded TGDs:

Theorem 5.4. *Fix $r \geq 1$. Let $\mathbb{O}$ be a recursively enumerable class of OMQs from $(\mathbb{G}, \mathbb{UCQ})$ over a schema of arity r, and, for each $k \geq 1$, $\mathbb{O} \not\subseteq (\mathbb{G}, \mathbb{UCQ})_k^{\equiv}$. Then,* p-OMQ-Evaluation($\mathbb{O}$) *is* W[1]*-hard.*

In view of Proposition 3.2, which states that, for each $k \geq 1$, OMQ-Evaluation($\mathbb{G}, \mathbb{CQ}_k$) is ExpTime-hard, it is not surprising that Theorem 5.3 does not state a pure tractability result. One may think that the above result can be easily obtained by using Grohe's construction underlying the fpt-reduction from p-Clique that establishes the lower bound of Theorem 4.1. However, the fact that the ontology can introduce additional relations that are not part of the data schema causes serious challenges. A detailed sketch of our proof is presented in Section 6, while full details are in the appendix.

Constraint Query Specifications. We now turn our attention to CQSs based on guarded TGDs. Interestingly, there is a strong connection between CQSs and OMQs with full data schema, which allows us to transfer results from the OMQ to the CQS setting. Recall that a full data schema consists of all the predicates occurring in the ontology and the UCQ. Note that we can naturally convert a CQS $S = (\Sigma, q)$ over a schema $\mathbf{S}$ into the OMQ $(\mathbf{S}, \Sigma, q)$ that has full data schema, denoted $\mathsf{omq}(S)$. The following result relates the UCQ_k-equivalence of CQSs to the UCQ_k-equivalence of OMQs.

Proposition 5.5. *Consider a CQS $S \in (\mathbb{G}, \mathbb{UCQ})$ over $\mathbf{T}$. For each $k \geq \mathsf{ar}(\mathbf{T}) - 1$, the following are equivalent:*

(1) S is uniformly $\mathbb{UCQ}_k$-equivalent.
(2) $\mathsf{omq}(S)$ is $\mathbb{UCQ}_k$-equivalent.

From Theorem 5.1 and Proposition 5.5, we get a 2ExpTime upper bound for deciding uniform $\mathbb{UCQ}_k$-equivalence for guarded CQSs, while the lower bound is inherited from [8]. As in Theorem 5.1, k should be greater than the arity of the schema, minus one.

Theorem 5.6. *Let S be a CQS from $(\mathbb{G}, \mathbb{UCQ})$ over a schema $\mathbf{T}$. For each $k \geq \mathsf{ar}(\mathbf{T}) - 1$, deciding whether S is uniformly $\mathbb{UCQ}_k$-equivalent is* 2ExpTime*-complete.*

Our main result concerning guarded CQSs shows that uniform $\mathbb{UCQ}_k$-equivalence characterizes tractability and fpt for classes of CQSs from $(\mathbb{G}, \mathbb{UCQ})$ over schemas of bounded arity.

Theorem 5.7 (**Main Result II**). *Fix $r \geq 1$. Let $\mathbb{O}$ be a recursively enumerable class of CQSs from $(\mathbb{G}, \mathbb{UCQ})$ over a schema of arity r. The following are equivalent, assuming* FPT $\neq$ W[1]*:*

(1) CQS-Evaluation($\mathbb{O}$) *is in* PTime.
(2) p-CQS-Evaluation($\mathbb{O}$) *is in* FPT.
(3) There is $k \geq 1$ such that $\mathbb{O} \subseteq (\mathbb{G}, \mathbb{UCQ})_k^{\equiv}$.

If either statement is false, then p-CQS-Evaluation($\mathbb{O}$) *is* W[1]*-hard.*

The fact that (3) implies (1) is shown in [8], while (1) implies (2) holds by definition. The interesting task is to show that (2) implies (3). To this end, we exploit the subsequent result which relates the evaluation of CQSs to the evaluation of OMQs. Given a class $\mathbb{O}$ of CQSs, we write $\mathsf{omq}(\mathbb{O})$ for the class of OMQs $\{\mathsf{omq}(S) \mid S \in \mathbb{O}\}$.

Proposition 5.8. *Consider a class $\mathbb{O}$ of CQSs from $(\mathbb{G}, \mathbb{UCQ})$. There exists an fpt-reduction from* p-OMQ-Evaluation($\mathsf{omq}(\mathbb{O})$) *to* p-CQS-Evaluation($\mathbb{O}$).

The reduction exploits the fact that guarded TGDs are *finitely controllable*, which means that OMQ-Evaluation($\mathbb{G}, \mathbb{UCQ}$) coincides with OMQ-Evaluation$_{\mathsf{fin}}$($\mathbb{G}, \mathbb{UCQ}$) that considers only *finite* models [4]. Note that the reduction does not extend to frontier-guarded TGDs since it requires the evaluation of TGD bodies to be fpt. Actually, no such reduction is possible for frontier-guarded TGDs (unless FPT = W[1]) since there exists a class $\mathbb{O}$ of CQSs from ($\mathbb{FG}, \mathbb{UCQ}_1$) such that p-OMQ-Evaluation(omq($\mathbb{O}$)) is W[1]-hard, while p-CQS-Evaluation($\mathbb{O}$) is in FPT: $\mathbb{O}$ consists of CQSs $(\Sigma, \exists x R(x))$, where Σ contains TGDs of the form $\varphi(x, \bar{y}) \rightarrow R(x)$, i.e., frontier-guarded TGDs with only one frontier variable. Now, by Theorem 5.4, Proposition 5.5, and Proposition 5.8, we get the following result, which establishes the direction (2) implies (3) of Theorem 5.7 (since we assume that FPT $\neq$ W[1]).

Theorem 5.9. *Fix $r \geq 1$. Let $\mathbb{O}$ be a recursively enumerable class of CQSs from ($\mathbb{G}, \mathbb{UCQ}$) over a schema of arity r, and, for each $k \geq 1$, $\mathbb{O} \not\subseteq (\mathbb{G}, \mathbb{UCQ})^{\equiv}_k$. Then,* p-CQS-Evaluation($\mathbb{O}$) *is* W[1]*-hard.*

5.2 The Frontier-guarded Case

By item (2) of Proposition 3.3, the notion of (uniform) $\mathbb{UCQ}_k$-equivalence does not provide a characterization of fpt for classes of OMQs from ($\mathbb{FG}, \mathbb{UCQ}$). Such a characterization result for OMQs based on frontier-guarded TGDs via an alternative notion of "being equivalent to an OMQ of low treewidth" remains an interesting open problem. The situation is different for querying in the presence of constraints. In fact, uniform $\mathbb{UCQ}_k$-equivalence allows us to characterize tractability and fpt for classes of CQSs from ($\mathbb{FG}, \mathbb{UCQ}$) over schemas of bounded arity r provided that the number of atoms in the head of TGDs is bounded by an integer m. The latter is required in our proof for reasons that are explained in Section 7.

The first question is whether uniform $\mathbb{UCQ}_k$-equivalence for CQSs based on frontier-guarded TGDs, with at most $m \geq 1$ head atoms, over schemas of arity $r \geq 1$, is decidable. We write $\mathbb{FG}_m$ for the class of frontier-guarded TGDs with at most m head atoms.

Theorem 5.10. *Fix $r, m \geq 1$. Let S be a CQS from ($\mathbb{FG}_m, \mathbb{UCQ}$) over a schema of arity r. For each $k \geq (r \cdot m - 1)$, deciding whether S is uniformly $\mathbb{UCQ}_k$-equivalent is* 2ExpTime*-complete.*

The above result exploits a characterization, analogous to Proposition 5.2, of when a CQS from ($\mathbb{FG}_m, \mathbb{UCQ}$) over a schema of arity r is uniformly $\mathbb{UCQ}_k$-equivalent via a suitable notion of $\mathbb{UCQ}_k$-approximation. The fact that we consider schemas of arity r, and frontier-guarded TGDs with at most m atoms in the head, allows us to show that chasing a database of treewidth k, with $k \geq r \cdot m - 1$, will lead to an instance that has treewidth k. This in turn allows us to adopt a notion of $\mathbb{UCQ}_k$-approximation that is significantly simpler than the one for guarded OMQs in Proposition 5.2. A *contraction* of a CQ q is a CQ obtained from q by identifying variables. When an answer variable x is identified with a non-answer variable y, the resulting variable is x, while the identification of two answer variables is not allowed. The $\mathbb{UCQ}_k$-approximation of a CQS $S = (\Sigma, q) \in (\mathbb{FG}, \mathbb{UCQ})$ is the CQS $S^a_k = (\Sigma, q^a_k)$, where q^a_k is the UCQ that consists of all contractions of a CQ from q that belong to $\mathbb{CQ}_k$. We can then show the following:

Proposition 5.11. *Fix $r, m \geq 1, k \geq r \cdot m - 1$. Let S be a CQS from ($\mathbb{FG}_m, \mathbb{UCQ}$) over a shema of arity r. The following are equivalent:*

(1) S is uniformly $\mathbb{UCQ}_k$-equivalent.
(2) $S \equiv S^a_k$.

From the above result, we get a procedure for deciding uniform $\mathbb{UCQ}_k$-equivalence: compute the $\mathbb{UCQ}_k$-approximation $S^a_k = (\Sigma, q^a_k)$ of $S = (\Sigma, q)$, and accept if $S \subseteq S^a_k$ (the other direction holds by construction); otherwise, reject. Clearly, S^a_k can be computed in exponential time, while q^a_k consists, in general, of exponentially many CQs, each of polynomial size. By Proposition 4.5, $S \subseteq S^a_k$ iff for each CQ $p(\bar{x}) \in q$, there exists $p' \in q^a_k$ such that $\bar{x} \in p'(\mathsf{chase}(p, \Sigma))$. Moreover, by Proposition 3.1 and item (1) of Proposition 3.2, we get that checking whether $\bar{x} \in p'(\mathsf{chase}(p, \Sigma))$ is feasible in double exponential time. Summing up, to check whether $S \subseteq S^a_k$ we need to perform exponentially many checks, while each check is feasible in double exponential time. This leads to the desired 2ExpTime upper bound, while a matching lower bound is inherited from [8].

Our main result concerning frontier-guarded CQSs follows:

Theorem 5.12 (**Main Result III**). *Fix $r, m \geq 1$. Let $\mathbb{O}$ be a recursively enumerable class of CQSs from ($\mathbb{FG}_m, \mathbb{UCQ}$) over a schema of arity r. The following are equivalent, assuming* FPT $\neq$ W[1]*:*

(1) CQS-Evaluation($\mathbb{O}$) *is in* PTime*.*
(2) p-CQS-Evaluation($\mathbb{O}$) *is in* FPT*.*
(3) There is $k \geq 1$ such that $\mathbb{O} \subseteq (\mathbb{FG}_m, \mathbb{UCQ})^{\equiv}_k$.

If either statement is false, then p-CQS-Evaluation($\mathbb{O}$) *is* W[1]*-hard.*

As for Theorem 5.7, the fact that (3) implies (1) is shown in [8], while (1) implies (2) holds by definition. Thus, the non-trivial task is to show that (2) implies (3). To this end, since we assume that FPT $\neq$ W[1], it suffices to show the following, which is our main technical result on CQSs based on frontier-guarded TGDs:

Theorem 5.13. *Fix $r, m \geq 1$. Let $\mathbb{O}$ be a recursively enumerable class of CQSs from ($\mathbb{FG}_m, \mathbb{UCQ}$) over a schema of arity r, and, for each $k \geq 1$, $\mathbb{O} \not\subseteq (\mathbb{FG}, \mathbb{UCQ})^{\equiv}_k$. Then,* p-CQS-Evaluation($\mathbb{O}$) *is* W[1]*-hard.*

Unlike Theorem 5.9, the above result cannot benefit from results on OMQs since (uniform) $\mathbb{UCQ}_k$-equivalence does not characterize fpt for OMQs based on frontier-guarded TGDs. It is shown via a reduction from p-Clique by following the same approach as for the lower bound of Grohe's Theorem, and exploiting the fact that frontier-guarded TGDs are finitely controllable. Let us stress, however, that our fpt-reduction is not a merely adaptation of Grohe's reduction. Unlike Grohe, we are more constrained in defining the right database as it must satisfy the given set of constraints. Moreover, the useful notion of core of a CQ, which was crucial for Grohe's proof, cannot be directly used in the presence of constraints.

The Rest of the Paper. We proceed to give some details on how our main lower bounds are shown; the full proofs are deferred to the appendix. Moreover, we defer to the appendix more details about the proofs of the results on the complexity of deciding $\mathbb{UCQ}_k$-equivalence, and of the upper bounds stated in our main results. The lower bounds for guarded OMQs and CQS, i.e., Theorems 5.4 and 5.9, respectively, are discussed in Section 6, while the lower bound for frontier-guarded CQSs, i.e., Theorem 5.13, in Section 7.

6 GUARDEDNESS

6.1 Ontology-mediated Queries

We provide a proof sketch of Theorem 5.4, illustrating only the main ideas and abstracting away many technical details. We rely on a result by Grohe, stated here in a form tailored towards our proof. For $k, \ell \geq 1$, the *$k \times \ell$-grid* is the graph with vertex set $\{(i, j) \mid 1 \leq i \leq k \text{ and } 1 \leq j \leq \ell\}$, and an edge between (i, j) and (i', j') iff $|i - i'| + |j - j'| = 1$. A *minor* of an undirected graph is defined in the usual way, see, e.g., [26]. When k is understood from the context, we use K to denote $\binom{k}{2}$. We say that a database D is *connected* if the graph G^D is connected. A constant $a \in \mathsf{dom}(D)$ is *isolated in D* if there is only a single atom $R(\bar{a}) \in D$ with $a \in \bar{a}$.

Theorem 6.1 (Grohe). *Given an undirected graph G, a $k \geq 1$, a connected* S*-database D, and a set $A \subseteq \mathsf{dom}(D)$ such that the restriction $G^D_{|_A}$ of the Gaifman graph of D to vertices A contains the $k \times K$-grid as a minor, one can construct in time $f(k) \cdot \mathsf{poly}(||G||, ||D||)$ an* S*-database D_G with $\mathsf{dom}(D) \setminus A \subseteq \mathsf{dom}(D_G)$ such that:*

1. *there is a surjective homomorphism h_0 from D_G to D that is the identity on $\mathsf{dom}(D) \setminus A$,*
2. *G contains a k-clique iff there is a homomorphism h from D to D_G such that h is the identity on $\mathsf{dom}(D) \setminus A$ and $h_0(h(\cdot))$ is the identity, and*
3. *if $R(a_1, \ldots, a_n) \in D$, $h_0(b_i) = a_i$ for each $i \in [n]$, and $\{b_i \mid a_i$ non-isolated in $D\}$ is a clique in the Gaifman graph of D_G, then D_G contains an atom $R(c_1, \ldots, c_n)$ where $c_i = b_i$ if a_i is non-isolated in D and $h_0(c_i) = a_i$ for $i \in [n]$.*

The condition in item (3) is not considered in [26], but can easily be verified to hold. It can be viewed as an ontoness requirement for the homomorphism h_0 from item (1) as it says that for certain atoms $R(\bar{a}) \in D$, we must find certain atoms $R(\bar{c}) \in D_G$ with $h_0(\bar{c}) = \bar{a}$. The set A is not present in Grohe's construction (that is, he considers the case $A = \mathsf{dom}(D)$), but we show in the appendix that it can be taken into account by a minor change in the construction.

For the proof of Theorem 5.4, we are given a recursively enumerable class of OMQs $\mathbb{O} \subseteq (\mathbb{G}, \mathbb{UCQ})$ such that $\mathbb{O} \not\subseteq (\mathbb{G}, \mathbb{UCQ})^{\equiv}_k$ for all $k \geq 1$. As a preliminary, we show that for every OMQ $Q \in \mathbb{O}$, we can effectively find an equivalent OMQ $Q' \in (\mathbb{G} \cap \mathbb{FULL}, \mathbb{UCQ})$ over the same extended schema, where $\mathbb{FULL}$ is the class of *full* TGDs, i.e., TGDs without existentially quantified variables. In other words, we can rewrite away the existential quantifiers in TGD heads. Although Q' need not be in $\mathbb{O}$, we can switch between Q and Q' in the proof whenever we only care about Q's semantics. To simplify this proof sketch, assume that no rewriting is needed as all ontologies in $\mathbb{O}$ are already from $\mathbb{G} \cap \mathbb{FULL}$, and assume that $\mathbb{O}$ contains only Boolean OMQs where the actual queries are CQs.

The proof is by fpt-reduction from p-Clique, a W[1]-hard problem. Assume that G is an undirected graph and $k \geq 1$ a clique size. We aim to construct a database D_G and find an OMQ $Q \in \mathbb{O}$ where

($*$) G has a k-clique iff $D_G \models Q$.

We first describe how to find Q. By Robertson and Seymour's Excluded Grid Theorem, there exists an ℓ such that every graph of treewidth exceeding ℓ contains the $k \times K$-grid as a minor [34]. We may assume, w.l.o.g., that ℓ exceeds the fixed arity r of relation symbols that occur in $\mathbb{O}$. Since $\mathbb{O} \not\subseteq (\mathbb{G}, \mathbb{UCQ})^{\equiv}_\ell$, there is an OMQ $Q = (\mathbf{S}, \Sigma, q)$ from $\mathbb{O}$ that is not in $(\mathbb{G}, \mathbb{UCQ})^{\equiv}_\ell$, and we can find Q by enumerating $\mathbb{O}$ and relying on Theorem 5.1. It thus remains to construct D_G. In the special case where Σ is empty and $\mathbf{S}$ is full, this can be done by replacing q with its core and applying Theorem 6.1, using $D[q]$ for D. Then, item (2) of that theorem corresponds to ($*$).

In the general case, however, we cannot use $D[q]$ for D since $D[q]$ might use relation symbols that are not in $\mathbf{S}$ and thus so would the resulting D_G. This, in fact, is what makes our proof rather delicate and difficult. The general idea is to replace $D[q]$ with an $\mathbf{S}$-database D_0 such that $D_0 \models Q$ and $(D_0)^u_\ell \not\models Q$, where $(D_0)^u_\ell$ denotes the unraveling of D_0 into a structure of treewidth ℓ. Ideally, we would like to apply the Grohe construction to D_0 instead of $D[q]$. However, we of course still have to attain ($*$), which is formulated in terms of Q (and thus, in terms of Σ and q) rather than D_0. This means that we need more from D_0 than just the stated properties and, in fact, we want $\mathsf{chase}(D_0, \Sigma)$ and q to be structurally as similar as possible. The following example illustrates a structural dissimilarity.

Example 6.2. Assume that Σ is empty, that $\mathbf{S}$ is full, and that q is the 3×4-grid, that is,

$$\bigwedge_{i \in \{1,2,3\}, j \in \{1,2,3\}} X(x_{i,j}, x_{i+1,j}) \wedge \bigwedge_{i \in \{1,2,3,4\}, j \in \{1,2\}} Y(x_{i,j}, x_{i,j+1}).$$

Moreover, let D_0 be the following database, the 3×3-grid augmented with reflexive X-loops in the rightmost column:

$$\{X(a_{i,j}, a_{i+1,j}) \mid i \in \{1, 2\}, j \in \{1, 2, 3\}\} \cup$$
$$\{Y(a_{i,j}, a_{i,j+1}) \mid i \in \{1, 2, 3\}, j \in \{1, 2\}\} \cup$$
$$\{X(a_{3,j}, a_{3,j}) \mid j \in \{1, 2, 3\}\}.$$

Clearly, $D_0 \models Q$ and $(D_0)^u_2 \not\models Q$.

Example 6.2 is a particularly simple case: the ontology is empty, the data schema is full, and the query is a core. We could in fact choose $D_0 = D[q]$. In general, however, it might not be possible to make D_0 and $D[q]$ isomorphic, and there might not even be some D_0 such that q maps injectively to D_0. What we do is choosing D_0 such that the following holds, where $D \models^{\text{io}} p$ means that $D \models p$ and all homomorphisms from p to D are injective ('io' stands for 'injectively only'):

($**$) if $\mathsf{chase}(D_0, \Sigma) \models^{\text{io}} q_c$, where q_c is a contraction of q, then there is no $\mathbf{S}$-database D and contraction q'_c of q such that D homorphically maps to D_0, $\mathsf{chase}(D, \Sigma) \models^{\text{io}} q'_c$, and $q_c \neq q'_c$ is a contraction of q'_c.

In Example 6.2, D_0 does not satisfy this condition as witnessed by choosing $D = D[q]$, where q_c is the contraction of q that identifies $x_{3,j}$ with $x_{4,j}$ for $j \in \{1, 2, 3\}$, and $q'_c = q$. At this point, $\mathsf{chase}(D_0, \Sigma)$ is still not as closely related to q as we need it to be. In fact, D_0 might contain parts that are superfluous for $D_0 \models Q$, and parts in which the atoms are unnecessarily entangled with each other. The following example illustrates the latter.

Example 6.3. Assume that q takes the form of an $n \times m$-grid that uses the binary relations X and Y. Let $\mathbf{S} = \{X', Y'\}$ and:

$$\Sigma = \{X'(x, y, z) \to X(x, y),\ Y'(x, y, z) \to Y(x, y)\}.$$

Further, let D_0 be the database

$$\{X'(a_{i,j}, a_{i,j+1}, b) \mid 1 \leq i \leq m \text{ and } 1 \leq j < n\} \cup$$
$$\{Y'(a_{i,j}, a_{i+1,j}, b) \mid 1 \leq i < m \text{ and } 1 \leq j \leq n\}.$$

Clearly, $D_0 \models Q$. However, the following 'untangled' homomorphic preimage D_1 of D_0 also satisfies $D_1 \models Q$, and for our proof it is much preferable to work with D_1 rather than with D_0:

$$\{X'(a_{i,j}, a_{i,j+1}, b_{ij}) \mid 1 \leq i \leq m \text{ and } 1 \leq j < n\} \cup$$
$$\{Y'(a_{i,j}, a_{i+1,j}, b'_{ij}) \mid 1 \leq i < m \text{ and } 1 \leq j \leq n\}.$$

Again, the example illustrates only a very simple case. To relate D_0 and q even closer, we apply to D_0 a 'diversification' construction that replaces each atom $R(\bar{a}) \in D_0$ with a (potentially empty) set of atoms $R(\bar{a}_1), \ldots, R(\bar{a}_n)$, where each $R(\bar{a}_i)$ can be obtained from $R(\bar{a})$ by replacing some constants with fresh isolated constants, and then attaching to the new atoms a finite initial piece of the guarded unraveling of D_0 that starts at $R(\bar{a})$. The latter part is necessary to not lose entailments of atoms through the ontology Σ; such entailments are preserved by guarded unraveling, which like the introduction of fresh constants is part of the 'untangling'. This diversification operation is made precise in the appendix. We use as D_G the database obtained by applying the Grohe construction to the result D_1 of diversifying D_0 in a maximal way such that $D_1 \models Q$. It then remains to show that $(*)$ holds.

The "only if" direction is rather straightforward. For the "if" direction, assume that $D_G \models Q$, that is, $\mathsf{chase}(D_G, \Sigma) \models q$. Thus, there is a contraction q'_c of q with $\mathsf{chase}(D_G, \Sigma) \models^{\text{io}} q'_c$. The composition g of a witnessing homomorphism with h_0 shows $\mathsf{chase}(D, \Sigma) \models q$, and thus there is a contraction q_c of q with $\mathsf{chase}(D, \Sigma) \models^{\text{io}} q_c$. Property $(**)$ is preserved by diversification, and thus by putting in D_1 for D_0 and D_G for D, we obtain $q_c = q'_c$. It follows that g must be injective and thus, so is h_0. We can also show that h_0 is 'sufficiently surjective and onto' so that from the inverse of h_0, we can construct a homomorphism h from D_1 to D_G that satisfies the conditions from Theorem 6.1, and then apply that theorem to conclude that G contains a k-clique, as needed.

We remark that our proof is considerably more involved than the one in [7], where ontologies are formulated in the description logic $\mathcal{ELHI}$, essentially a fragment of $\mathbb{G}$. In particular, the presence of relations of arity larger than two makes it necessary to use diversifications, isolated constants, and condition (iii) from Theorem 6.1, while none of this is necessary for the $\mathcal{ELHI}$ case.

6.2 Constraint Query Specifications

We now discuss the fpt-reduction of Proposition 5.8, which is underlying Theorem 5.9. Recall that this reduction relies on the fact that guarded TGDs are finitely controllable. Recall also that the notion of finite controllability is crucial for establishing our main technical result on CQSs based on frontier-guarded TGDs (Theorem 5.13), for which more details are given in the next section. Let us then give some useful details around finite controllability.

Finite Controllability. Given an $\mathbf{S}$-database D, and a set Σ of TGDs over $\mathbf{S}$, we write $\mathsf{fmods}(D, \Sigma)$ for the set of all *finite* models of D and Σ. Finite controllability guarantees that, for computing the set of tuples that are answers to a UCQ q over every finite model of D and Σ, i.e., the set $\bigcap_{M \in \mathsf{fmods}(D,\Sigma)} q(M)$, it suffices to evaluate q over the (possibly infinite) instance $\mathsf{chase}(D, \Sigma)$.

Definition 6.4 (**Finite Controllability**). A class $\mathbb{C}$ of TGDs is *finitely controllable* if, for every $\mathbf{S}$-database D, set $\Sigma \in \mathbb{C}$ of TGDs over $\mathbf{S}$, and UCQ q over $\mathbf{S}$, $q(\mathsf{chase}(D, \Sigma)) = \bigcap_{M \in \mathsf{fmods}(D,\Sigma)} q(M)$. ■

Actually, as we shall see, in our proofs we rely on a seemingly stronger property than finite controllability. Roughly speaking, this property states that, once we fix the number of variables that can appear in UCQs, let say $n \geq 0$, given a database D, and a set Σ of TGDs, there exists a finite model M of D and Σ, which depends on n, that allows us to compute the set of tuples $q(\mathsf{chase}(D, \Sigma))$, i.e., $q(\mathsf{chase}(D, \Sigma)) = q(M)$, no matter what the UCQ looks like.

Definition 6.5 (**Strong Finite Controllability**). Consider a class $\mathbb{C}$ of TGDs. We say that $\mathbb{C}$ is *strongly finitely controllable* if, for every $\mathbf{S}$-database D, set $\Sigma \in \mathbb{C}$ of TGDs over $\mathbf{S}$, and integer $n \geq 0$, there exists an instance $M(D, \Sigma, n) \in \mathsf{fmods}(D, \Sigma)$ such that, for each UCQ q that has at most n variables, $q(\mathsf{chase}(D, \Sigma)) = q(M(D, \Sigma, n))$. If there is a computable function that takes as input D, Σ, and n, and outputs $M(D, \Sigma, n)$, then we say that *a finite witness is realizable.* ■

The next technical result, which is of independent interest, shows that finite and strong finite controllability are equivalent properties.

Lemma 6.6. *Consider a class $\mathbb{C}$ of TGDs. It holds that $\mathbb{C}$ is finitely controllable iff $\mathbb{C}$ is strongly finitely controllable.*

By using the above lemma and results on the *guarded negation* fragment of first-order logic (GNFO) from [5], we can show that:

Theorem 6.7. *The class $\mathbb{FG}$ is strongly finitely controllable. Moreover, finite witnesses are realizable.*

In particular, every satisfiable GNFO sentence has a finite model. This allows us to show that the class of frontier-guarded TGDs is finitely controllable, and thus, by Lemma 6.6, it is also strongly finitely controllable. Moreover, we know that if a GNFO sentence φ has a finite model, then it has a finite model of size $2^{2^{||\varphi||^{O(1)}}}$. This allows us to show, in addition, that finite witnesses are realizable.

The FPT-Reduction. Let us now come back to Proposition 5.8. We want to show the following: given a class of CQSs $\mathbb{O}$ from $(\mathbb{G}, \mathbb{UCQ})$, there exists an fpt-reduction from p-OMQ-Evaluation($\mathsf{omq}(\mathbb{O})$) to p-CQS-Evaluation($\mathbb{O}$). Consider $Q = (\mathbf{S}, \Sigma, q(\bar{x}))$ from $\mathsf{omq}(\mathbb{O})$, an $\mathbf{S}$-database D, and a tuple $\bar{c} \in \mathsf{dom}(D)^{|\bar{x}|}$. We are going to construct an $\mathbf{S}$-database D^* such that $D^* \models \Sigma$, and $\bar{c} \in Q(D)$ iff $\bar{c} \in q(D^*)$, or, equivalently (due to Proposition 2.2), $\bar{c} \in \mathsf{chase}(D, \Sigma)$ iff $\bar{c} \in q(D^*)$. We first define D^+ as the database

$$D \cup \{R(\bar{a}) \in \mathsf{chase}(D, \Sigma) \mid \bar{a} \subseteq \mathsf{dom}(D)\}.$$

Let A be the family of all maximal tuples $\bar{a}$ over $\mathsf{dom}(D)$ that are guarded in D^+, i.e., there is an atom $R(\bar{b}) \in D^+$ such that $\bar{a} \subseteq \bar{b}$. Fix an arbitrary tuple $\bar{a} \in A$. Since, by Theorem 6.7, the class $\mathbb{G}$ is strongly finitely controllable, and also finite witnesses are realizable, we can compute an instance $M(D^+_{|\bar{a}}, \Sigma, n) \in \mathsf{fmods}(D^+_{|\bar{a}}, \Sigma)$, where n is the number of variables in q, such that for each CQ q' of arity $|\bar{a}|$ with at most n variables, it holds that

$$(*)\ \ \bar{a} \in q'(M(D^+_{|\bar{a}}, \Sigma, n)) \implies \bar{a} \in q'(\mathsf{chase}(D^+_{|\bar{a}}, \Sigma)).$$

W.l.o.g., we assume that $\mathsf{dom}(M(D^+_{|\bar{a}}, \Sigma, n)) \cap \mathsf{dom}(M(D^+_{|\bar{b}}, \Sigma, n)) \subseteq \mathsf{dom}(D)$, for every two distinct tuples $\bar{a}, \bar{b} \in A$. The database D^* is

$$D^+ \cup \bigcup_{\bar{a} \in A} M(D^+_{|\bar{a}}, \Sigma, n).$$

The next lemma shows that D^* is the desired database.

Lemma 6.8. *It holds that:*

(1) $D^* \models \Sigma$.
(2) $\bar{c} \in \mathsf{chase}(D, \Sigma)$ *iff* $\bar{c} \in q(D^*)$.
(3) There exists a computable function $f : \mathbb{N} \to \mathbb{N}$ *such that* D^* *can be constructed in time* $||D||^{O(1)} \cdot f(||Q||)$.

Item (1) can be shown by exploiting the fact that Σ is guarded. Concerning item (2), the ($\Rightarrow$) direction is shown by exploiting the universality of the chase (Proposition 2.2), while the ($\Leftarrow$) direction by using the implication ($*$). Finally, item (3) relies on the fact that the database D^+ can be constructed in time $||D||^{O(1)} \cdot g(||Q||)$ for some computable function $g : \mathbb{N} \to \mathbb{N}$, and also on the observation that the cardinality of A, as well as the size of $D_{|\bar{a}}$ for some $\bar{a} \in A$, do not depend on D. The full proof can be found in the appendix.

7 FRONTIER-GUARDEDNESS

In this final section, we focus on the proof of Theorem 5.13. Consider a recursively enumerable class $\mathbb{O}$ of CQSs from $(\mathbb{FG}_m, \mathbb{UCQ})$ over a schema of arity r such that, for each $k \geq 1$, $\mathbb{O} \not\subseteq (\mathbb{FG}_m, \mathbb{UCQ})^{\equiv}_k$. Our goal is to show that CQS-Evaluation($\mathbb{O}$) is W[1]-hard. Note that, unlike in the proof of Theorem 5.4, we cannot eliminate the existential quantifiers. The reason is that in the OMQ case we only need an equivalent OMQ while in the CQS case we would have to replace the set of constraints Σ from $\mathbb{FG}$ with an equivalent set Σ' from $\mathbb{FG} \cap \mathbb{FULL}$, but clearly such a Σ' need not exist. As in the proof of Grohe's Theorem, we provide an fpt-reduction from p-Clique (in fact, a restricted version of it as explained later). For the sake of clarity, here we discuss the case where $\mathbb{O}$ consists only of CQSs $Q = (\Sigma, q)$ where q is a Boolean CQ whose Gaifman graph is connected. The extension to non-Boolean UCQs consisting of non-connected CQs is treated in the appendix.

A Variation of Grohe's Database. For technical reasons that we clarify later, we do not use Grohe's database used in [26] to establish Theorem 4.1, nor the one that we used for showing Theorem 5.4. Instead, we use a subset of Grohe's database that satisfies several good properties summarized in the next result.

Theorem 7.1 (Grohe). *Given an undirected graph* $G = (V, E)$, $k \geq 1$, *databases* D, D' *with* $D \subseteq D'$, *and a set* $A \subseteq \mathsf{dom}(D)$ *such that the restriction* $G^D_{|A}$ *of the Gaifman graph of* D *to vertices* A *contains the* $k \times K$*-grid as a minor, one can construct in time* $f(k) \cdot \mathsf{poly}(||G||, ||D'||)$ *a database* $D^* = D^*(G, D, D', A)$ *such that:*

(1) there is a surjective homomorphism h_0 *from* D^* *to* D',
(2) G *contains a* k*-clique iff there is a homomorphism* h *from* D *to* D^* *such that* $h_0(h(\cdot))$ *is the identity on* A, *and*
(3) if $D' \models \Sigma$, *and every clique of size at most* $3 \cdot r$ *in* G *is contained in a clique of size* $3 \cdot r \cdot m$, *then* $D^* \models \Sigma$.

The condition in item (3) is not considered by Grohe in [26], but can easily be verified to hold for our modified version of his database. It is crucial for our proof as we need to ensure that the database that the fpt-reduction constructs satisfies the given set Σ of frontier-guarded TGDs.

A Crucial Lemma. Our proof borrows several ideas and techniques from the proof of Grohe's Theorem. However, the adaptation of such techniques is non-trivial for a couple of reasons. First, we cannot construct an arbitrary database, but need to construct one that satisfies the given set Σ of frontier-guarded TGDs. Second, the notion of core of a CQ, which is crucial for Grohe's proof, cannot be directly used in our context. Recall that, for $k \geq 1$, a CQ q belongs to $\mathbb{CQ}^{\equiv}_k$ iff its core is in $\mathbb{CQ}_k$ [20]. This allows Grohe to work with the core of q instead of q itself, and thus, exploit the property that a homomorphism from the core of q to itself is injective. This is far from being true in the presence of constraints. Instead, we need a technical result, which, intuitively speaking, states that some subsets of our CQs behave like cores for the sake of our proof.

Lemma 7.2. *Fix* $\ell \geq r \cdot m$. *There exists a computable function that takes as input a CQS* $S = (\Sigma, q) \in (\mathbb{FG}_m, \mathbb{CQ})$ *that is not uniformly* $\mathbb{CQ}_\ell$*-equivalent, and outputs two CQs* p *and* p', *and a subset* X *of the variables of* p, *such that the following hold:*

(1) $q \equiv_\Sigma p$.
(2) $D[p'] \models \Sigma$.
(3) $D[p] \subseteq D[p']$.
(4) $h(X) = X$, *for every homomorphism* h *from* p *to* p'.
(5) The treewidth of $G^p_{|X}$ *is larger than* ℓ.

The proof of the above lemma can be found in the appendix. Let us now briefly comment on the reason why we have to bound the number of head atoms in frontier-guarded TGDs for our proof to work. In very rough terms, this is because the way in which p' is constructed is by generating from p the finite model $M = M(D[p], \Sigma, n)$, for some $n \geq 0$, that is obtained from Definition 6.5 given the strong finite controllability of $\mathbb{FG}$, as established in Theorem 6.7. A property of this construction that is crucial for the proof of Lemma 7.2 is that the treewidth of M is not larger than that of $D[p]$. The only way in which we can ensure this property to hold is by fixing the number of atoms in TGD heads.

Remarkably, this additional bound on the number of head atoms is not needed for obtaining Theorem 5.7. This is because there exists an fpt-reduction from the evaluation of CQSs from $(\mathbb{G}, \mathbb{UCQ})$ to CQS from the same class in which there are are no existential quantifiers in TGD heads – this is implicit in our proofs. When such a reduction is possible, the CQ p' can be obtained in a simpler way: it suffices to chase p using Σ, with p being the CQ with a minimal number of atoms among all the CQs that are equivalent to q w.r.t. Σ. Notice that no bound on the number of head atoms in the TGDs of Σ is required in this case. Unfortunately, when $\Sigma \in \mathbb{FG}$, a reduction like the one described above is not possible. This prevents us from applying the previous idea since $\mathsf{chase}(p, \Sigma)$ may be infinite.

The FPT-Reduction. We now proceed to explain how Lemma 7.2 is applied in order to prove Theorem 5.13. Let (G, k) be an instance of p-Clique. It is easy to see that we can assume, w.l.o.g., that every clique of size at most $3 \cdot r$ in G is contained in a clique of size $3 \cdot r \cdot m$. From the Excluded Grid Theorem [34], there is a computable integer ℓ such that every simple graph G of treewidth at least ℓ contains a $(k \times K)$-grid as a minor. By hypothesis on the class $\mathbb{O}$, there exists a CQS $S = (\Sigma, q)$ from $\mathbb{O}$ such that $S \notin (\mathbb{FG}_m, \mathbb{CQ})^{\equiv}_\ell$. We can assume, w.l.o.g., that $\ell \geq r \cdot m$. We build from (G, k) the tuple (D^*, Σ, q), where D^* is a database defined as follows. Since, by assumption, $\ell \geq r \cdot m$, it is possible to compute from q a CQ p, a subset X of the variables of p, and a CQ p', that satisfy the properties stated in Lemma 7.2. In particular, the treewidth of $G^p_{|X}$ is at least ℓ, and hence, $G^p_{|X}$ contains the $(k \times K)$-grid as a minor. We then define D^*

as $D^*(G, D[p], D[p'], X)$, where $D(G, D[p], D[p'], X)$ is the database that is defined in Theorem 7.1 for $D = D[p]$, $D' = D[p']$, and $A = X$.

It remains to show that the above is indeed an fpt-reduction from p-Clique to CQS-Evaluation($\mathbb{O}$). To this end, we need to show that:

Lemma 7.3. *The following statements hold:*

(1) $D^* \models \Sigma$.
(2) G *has a* k*-clique iff* $D^* \models q$.
(3) There are computable functions $f, g : \mathbb{N} \to \mathbb{N}$ *such that* (D, Σ, q) *can be constructed in time* $||G||^{O(1)} \cdot f(k)$ *and* $(||q|| + ||\Sigma||) \leq g(k)$.

The proof of (1) follows from the last item in Theorem 7.1. We now proceed to show item (2). Assume that G has a k-clique. Then, by Theorem 7.1, there is a homomorphism h from p to D^* such that $h_0(h(\cdot))$ is the identity. Hence $D^* \models p$. But $q \equiv_\Sigma p$, and thus, $D^* \models q$ since, by item (1), we know that $D^* \models \Sigma$. Conversely, assume that $D^* \models q$, and thus $D^* \models p$. Then, there is a homomorphism h from p to D^*. It follows that $h_0(h(\cdot))$ is a homomorphism from p to p'. Consequently, by Lemma 7.2, we obtain that $h(X) = X$. Hence, there must exist some $n \geq 0$ such that $g = h \circ (h_0 \circ h)^n$ is a homomorphism from p to D^* with $h_0(g(\cdot))$ being the identity on X. Then, G has a k-clique from item (2) of Theorem 7.1.

As for items (3) and (4), first notice that the CQS $S = (\Sigma, q)$ can be computed by simply enumerating the CQSs from $\mathbb{O}$ until we find S since, by Theorem 5.10, we can check whether $S \notin (\mathbb{FG}, \mathbb{CQ})^{\equiv}_{\ell}$. From q we can construct the CQs p and p', as well as the set of variables X, by applying Lemma 7.2. All these constructions depend only on k. Theorem 7.1 states, on the other hand, that it is possible to construct D in time $||G||^{O(1)} \cdot f'(k)$ for some computable function $f' : \mathbb{N} \to \mathbb{N}$. Putting all these together, we obtain the existence of computable functions $f, g : \mathbb{N} \to \mathbb{N}$ as needed.

'

8 RELATIONS OF UNBOUNDED ARITY

We take a first look at the case where the arity of relations is not bounded by a constant, focussing on linear TGDs and the OMQ language $(\mathbb{L}, \mathbb{UCQ})$. In contrast to the OMQ languages studied in previous sections, every OMQ from $(\mathbb{L}, \mathbb{UCQ})$ is rewritable into an equivalent UCQ. We are going to exploit this property in our analysis. For simplicity, we only treat Boolean OMQs in what follows, so without further mention all queries considered in this section are assumed to be Boolean.

We start with some preliminaries. A *hypergraph* is a pair $H = (V, E)$ with V a set of *nodes* and $E \subseteq 2^V \setminus \{\emptyset\}$ a set of *edges*. A *tree decomposition* of a hypergraph is defined like that of a graph in Section 2 except that the second condition now says: if $e \in E$, then $e \subseteq \chi(t)$ for some $t \in V_\delta$. The *treewidth* of a hypergraph is then defined in the expected way. Every CQ q gives rise to an associated hypergraph $H[q]$ whose vertices are the variables of q and which contains an edge $\mathbf{x}$ for every atom $R(\mathbf{x})$ of q.

Let $H = (V, E)$ be a hypergraph. A function $f : 2^V \to \mathbb{R}^+$ is *submodular* if $f(X) + f(Y) \geq f(X \cap Y) + f(X \cup Y)$. We only consider submodular functions f that satisfy $f(\emptyset) = 0$ and are *edge dominated*, that is, $f(e) \leq 1$ for all $e \in E$. The *submodular width* of H is the smallest k such that for every monotone submodular function f, there is a tree decomposition (T_δ, χ) of H, $T = (V_\delta, E_\delta)$ with $f(\chi(t)) \leq k$ for all $t \in V_\delta$. The submodular width of a CQ q is that of $H[q]$. A class of CQs $\mathbb{Q}$ *has bounded submodular width* if there exists a $k > 0$ such that the submodular width of each $q \in Q$ is at most k.

A CQ q is *self-join free* if no relation symbol occurs in more than one atom in q. The *exponential time hypothesis (ETH)* says that there is no algorithm for n-variable 3SAT that runs in time $2^{o(n)}$ [27]. It implies FPT $\neq$ W[1]. The following result can to some extent be viewed as a companion of Theorem 6.1 for the case of unbounded relation arity.

Theorem 8.1 (Marx). *Let* $\mathbb{Q}$ *be a recursively enumerable class of CQs such that for every* $q \in \mathbb{Q}$*, there is a self-join free* $q' \in \mathbb{Q}$ *with* $H[q] = H[q']$*. The following are equivalent unless ETH fails:*

(1) $\mathbb{Q}$ *can be evaluated in FPT;*
(2) $\mathbb{Q}$ *has bounded submodular width.*

The "(2) $\Rightarrow$ (1)*" direction also holds without the condition on* $\mathbb{Q}$ *and does not depend on ETH.*

This formulation is actually a slight strengthening of that of Marx who only demands that $\mathbb{Q}$ is closed under CQs that have the same hypergraph. Marx' proof, however, also establishes this slightly stronger version. Note that, in contrast to Grohe's Theorem, Theorem 8.1 does not speak about *equivalence* to CQs of bounded width. This is related to the assumption about self-join freeness: the hypergraph of every self-join free query is a core and informally, there is no gain in transitioning from a core to an equivalent query. It is not known whether Marx' result holds also without this assumption.

As a preparation for what follows, we first lift Marx' result to the UCQ case. We call a UCQ *canonical* if the CQs in it are mutually homomorphically incomparable. It is easy to see that every UCQ is equivalent to a canonical UCQ. A class of UCQs $\mathbb{Q}$ *has bounded submodular width* if there exists a k such that every CQ that occurs in some $q \in Q$ has submodular width at most k.

Theorem 8.2. *Let* $\mathbb{Q}$ *be a recursively enumerable class of canonical UCQs such that for every UCQ* $q \in \mathbb{Q}$ *and every CQ* p *in* q*, there is a UCQ* $q' \in \mathbb{Q}$ *and a self-join free CQ* p' *in* q' *with* $H[p] = H[p']$*. The following are equivalent unless ETH fails:*

(1) $\mathbb{Q}$ *can be evaluated in FPT;*
(2) $\mathbb{Q}$ *has bounded submodular width.*

The "(2) $\Rightarrow$ (1)*" direction also holds without the condition on* $\mathbb{Q}$ *and does not depend on ETH.*

Proof. The "(2) $\Rightarrow$ (1)" direction is trivial: one can simply evaluate each CQ in a given UCQ and consider the union of results.

As concerns the "(1) $\Rightarrow$ (2)" direction, assume that $\mathbb{Q}$ has unbounded submodular width. Then for every k, there is a UCQ q_k in $\mathbb{Q}$ and a CQ p_k in q_k such that p_k has submodular width higher than k. Furthermore, there must be some UCQ $\widehat{q}_k$ in $\mathbb{Q}$ and some CQ $\widehat{p}_k$ in $\widehat{q}_k$ such that $\widehat{p}_k$ is self-join free and has submodular width higher than k. Let $\mathbb{Q}'$ be the class of all CQs $\widehat{p}_k$, $k > 0$. Then, each CQ in $\mathbb{Q}'$ is self-join free, and thus $\mathbb{Q}'$ meets the prerequisites from Theorem 8.1.

As $\mathbb{Q}'$ does not have bounded submodular width, it follows from Theorem 8.1 that $\mathbb{Q}'$ cannot be evaluated in FPT. We show that the same holds for $\mathbb{Q}$ by reducing the evaluation of each CQ from $\mathbb{Q}'$

to the evaluation of some UCQ from $\mathbb{Q}$. Let $\widehat{p}_k$ be a CQ from $\mathbb{Q}'$, for some $k > 0$, and D be some database. By construction, there is a UCQ $\widehat{q}_k$ in $\mathbb{Q}$ which contains $\widehat{p}_k$ as a CQ and in which each two CQs are homomorphically incomparable. Then, for every CQ p' in $\widehat{q}_k$ with $p' \neq \widehat{p}_k$, $D[\widehat{p}_k] \not\models p'$ (due to the fact that $\widehat{p}_k$ and p' are homomorphically incomparable). Moreover, $D[\widehat{p}_k] \models \widehat{p}_k$. It follows that $D \models \widehat{p}_k$ iff $D \times D[\widehat{p}_k] \models q_k$. □

The assumption about self-join freeness also needs to be reflected in the intended classification of OMQs. It makes, however, little sense to adopt it verbatim: to obtain a self-join free CQ with the same hypergraph, one has to replace relation symbols in the query with fresh ones, but this breaks the connection to the ontology. We thus develop a more natural and mild condition. A class of OMQs $\mathbb{O} \subseteq (\mathbb{L}, \mathbb{UCQ})$ is *closed under fresh implicants* if whenever $(\Sigma, \mathbf{S}, q) \in \mathbb{O}$, $R \in \mathbf{S}$, and R' is a relation symbol of the same arity as R that does not occur in Σ or q, then $Q' = (\Sigma \cup \{R'(\mathbf{x}) \to R(\mathbf{x})\}, \mathbf{S} \cup \{R'\}, q) \in \mathbb{O}$ where $\mathbf{x}$ is a tuple of variables of suitable length without repetitions. We then also call R' a *fresh implicant* of R and say that Q' is *obtained from Q by adding a fresh implicant of R*.

The submodular width of an OMQ $(\Sigma, \mathbf{S}, q)$ is that of q. For $k > 0$, we use $(\text{linTGD}, \text{UCQ})_{\text{SMW}(k)}$ to denote the class of OMQs from (linTGD, UCQ) that are of submodular width at most k and $(\mathbb{L}, \mathbb{UCQ})^{\equiv}_{\text{SMW}(k)}$ to denote the class of OMQs $Q \in (\mathbb{L}, \mathbb{UCQ})$ that are equivalent to an OMQ from $(\text{linTGD}, \text{UCQ})_{\text{SMW}(k)}$. The main result in this section is the following.

Theorem 8.3. *For any recursively enumerable class $\mathbb{Q}$ of OMQs that is closed under fresh implicants, the following are equivalent unless ETH fails:*

(1) $\mathbb{Q}$ can be evaluated in FPT;
(2) $\mathbb{Q} \subseteq (\text{linTGD}, \text{UCQ})^{\equiv}_{\text{SMW}(k)}$ for some $k > 0$.

The "(2) $\Rightarrow$ (1)" direction also holds without fresh implicants and does not depend on ETH.

To prove Theorem 8.3, we exploit the fact that every OMQ $Q \in (\mathbb{L}, \mathbb{UCQ})$ can be rewritten into a UCQ q that is equivalent to Q [15]. Call a UCQ *core* if all CQs in it are cores. For a class of OMQs $\mathbb{Q} \subseteq (\mathbb{L}, \mathbb{UCQ})$, we use $\text{rew}(\mathbb{Q})$ to denote the set of all canonical core UCQs that are equivalent to some OMQ from $\mathbb{Q}$. The following is a central observation.

Lemma 8.4. *For every recursively enumerable $\mathbb{Q} \subseteq (\mathbb{L}, \mathbb{UCQ})$ that is closed under fresh implicants, UCQ $q \in \text{rew}(\mathbb{Q})$, and CQ p in q, there is a UCQ $q' \in \text{rew}(\mathbb{Q})$ and a self-join free CQ p' in q' such that $H[p] = H[p']$.*

Proof. Let $\mathbb{Q}$, q, and p be as in the lemma. Also, let $Q = (\Sigma, \mathbf{S}, q) \in \mathbb{Q}$ be an OMQ that is equivalent to q. As $\mathbb{Q}$ is closed under fresh implicants, there must be an OMQ $Q' \in \mathbb{Q}$ that is obtained from Q by adding fresh implicants and such that for every $R \in \mathbf{S}$, the number of fresh implicants of R in Q' is bounded from below by the maximum size of CQs in q. Let q' be the UCQ obtained from q by adding as disjuncts all CQs obtained from a CQ in q by replacing atoms $R(\mathbf{x})$ with $R'(\mathbf{x})$, R' a fresh implicant of R in Q'. We observe the following:

(1) q' is equivalent to Q';
(2) q' is canonical.

sketch the arguments

It follows from Points 1 and 2 abve that $q' \in \text{rew}(\mathbb{Q})$. Furthermore, by construction q' contains a CQ p that has been obtained from p by replacing each atom with one based on a different fresh implicant, and thus p' is self-join free and $H[p] = H[p']$, as required. □

We are now ready to prove Theorem 8.3.

"(1) $\Rightarrow$ (2)'. Assume that $\mathbb{Q}$ can be evaluated in FPT. Then so can $\text{rew}(\mathbb{Q})$ because for every $q \in \text{rew}(\mathbb{Q})$ we can effectively find a $Q \in \mathbb{Q}$ that is equivalent: enumerate all OMQs from $\mathbb{Q}$, construct a UCQ-rewriting q' of Q, and check whether $q \equiv q'$. By Lemma 8.4 and Theorem 8.2, the submodular width of the UCQs in $\text{rew}(\mathbb{Q})$ is bounded by some k. We can view each query from $\text{rew}(\mathbb{Q})$ as an OMQ from (linTGD, UCQ) in which the ontology happens to be empty. Consequently, $\mathbb{Q} \subseteq (\text{linTGD}, \text{UCQ})^{\equiv}_{\text{SMW}(k)}$.

"(2) $\Rightarrow$ (1)'. Assume that $\mathbb{Q} \subseteq (\text{linTGD}, \text{UCQ})^{\equiv}_{\text{SMW}(k)}$ for some $k > 0$. It suffices to argue that $\text{rew}(\mathbb{Q})$ can be evaluated in FPT as we can effectively convert every OMQ from $\mathbb{Q}$ into an equivalent one from $\text{rew}(\mathbb{Q})$.

By Lemma A.1 and Theorem 8.2, the class $(\text{linTGD}, \text{UCQ})_{\text{SMW}(k)}$ can be evaluated in FPT, and thus so can $\text{rew}((\text{linTGD}, \text{UCQ})_{\text{SMW}(k)})$. It is therefore enough to show that $\text{rew}(\mathbb{Q}) \subseteq \text{rew}((\text{linTGD}, \text{UCQ})_{\text{SMW}(k)})$.

Let $q \in \text{rew}(\mathbb{Q})$. Then there is a $Q \in \mathbb{Q}$ that is equivalent to q and a $Q_k \in (\text{linTGD}, \text{UCQ})_{\text{SMW}(k)}$ that is equivalent to Q and thus to q. It follows that $\text{rew}((\text{linTGD}, \text{UCQ})_{\text{SMW}(k)})$ contains a UCQ q_k that is equivalent to q. Both q and q_k consists of CQs that are homomorphically incomparable cores. It follows that for every CQ in q, there is an isomorphic one in q_k and vice versa. Thus, $q \in \text{rew}((\text{linTGD}, \text{UCQ})_{\text{SMW}(k)})$.

It seems worthwhile to point out that it follows from the above proof that if $\mathbb{Q} \subseteq (\text{linTGD}, \text{UCQ})^{\equiv}_{\text{SMW}(k)}$, then the submodular width of the CQs in (the UCQs in) $\text{rew}(Q)$ is also bounded by k. This of course depends on the fact that we assume all such CQs to be cores.

9 CONCLUSIONS AND FUTURE WORK

We have studied the limits of efficient query evaluation in the context of ontology-mediated querying and querying in the presence of constraints, focussing on guarded and frontier-guarded TGDs, and on UCQs as the actual queries. We have obtained novel efficiency characterizations in the spirit of Grohe's well-known characterization of the tractable classes of CQs (without constraints).

An interesting open problem that emerges from this work is whether our main result on guarded OMQs (Theorem 5.3) can be generalized to frontier-guarded TGDs. Recall that the notion of being equivalent to an OMQ of bounded treewdith is not enough for frontier-guarded TGDs. We would also like to investigate whether in our main result on CQSs based on frontier-guarded TGDs (Theorem 5.12) the bound on the number of head atoms can be dropped. Recall that our results, similarly to Grohe's characterization, assume that the underlying schema is of bounded arity. Establishing similar efficiency characterizations for schemas of unbounded arity

is a challenging open problem, where the more involved notion of submodular width, introduced by Marx [31], should be considered.

REFERENCES

[1] Serge Abiteboul, Richard Hull, and Victor Vianu. 1995. *Foundations of Databases*. Addison-Wesley.

[2] Jean-François Baget, Marie-Laure Mugnier, Sebastian Rudolph, and Michaël Thomazo. 2011. Walking the Complexity Lines for Generalized Guarded Existential Rules. In *IJCAI*. 712–717.

[3] Jean-François Baget, Michel Leclère, Marie-Laure Mugnier, and Eric Salvat. 2011. On rules with existential variables: Walking the decidability line. *Artif. Intell.* 175, 9-10 (2011), 1620–1654.

[4] Vince Bárány, Georg Gottlob, and Martin Otto. 2014. Querying the Guarded Fragment. *Logical Methods in Computer Science* 10, 2 (2014).

[5] Vince Bárány, Balder ten Cate, and Luc Segoufin. 2015. Guarded Negation. *J. ACM* 62, 3 (2015), 22:1–22:26.

[6] Pablo Barceló, Gerald Berger, and Andreas Pieris. 2018. Containment for Rule-Based Ontology-Mediated Queries. In *PODS*. 267–279.

[7] Pablo Barceló, Cristina Feier, Carsten Lutz, and Andreas Pieris. 2019. When is Ontology-Mediated Querying Efficient?. In *LICS*. 1–13.

[8] Pablo Barceló, Diego Figueira, Georg Gottlob, and Andreas Pieris. 2019. Semantic Optimization of Conjunctive Queries. Under submission, available at http://homepages.inf.ed.ac.uk/apieris/BFGP.pdf.

[9] Pablo Barceló, Georg Gottlob, and Andreas Pieris. 2016. Semantic Acyclicity Under Constraints. In *PODS*. 343–354.

[10] Meghyn Bienvenu, Peter Hansen, Carsten Lutz, and Frank Wolter. 2016. First Order-Rewritability and Containment of Conjunctive Queries in Horn Description Logics. In *IJCAI*. 965–971.

[11] Meghyn Bienvenu and Magdalena Ortiz. 2015. Ontology-Mediated Query Answering with Data-Tractable Description Logics. In *Reasoning Web*. 218–307.

[12] Meghyn Bienvenu, Balder ten Cate, Carsten Lutz, and Frank Wolter. 2014. Ontology-Based Data Access: A Study through Disjunctive Datalog, CSP, and MMSNP. *ACM Trans. Database Syst.* 39, 4 (2014), 33:1–33:44.

[13] Achim Blumensath. 2010. Guarded Second-Order Logic, Spanning Trees, and Network Flows. *Logical Methods in Computer Science* 6, 1 (2010).

[14] Andrea Calì, Georg Gottlob, and Michael Kifer. 2013. Taming the Infinite Chase: Query Answering under Expressive Relational Constraints. *J. Artif. Intell. Res.* 48 (2013), 115–174.

[15] Andrea Calì, Georg Gottlob, and Thomas Lukasiewicz. 2012. A general Datalog-based framework for tractable query answering over ontologies. *J. Web Sem.* 14 (2012), 57–83.

[16] Andrea Calì, Georg Gottlob, and Andreas Pieris. 2012. Towards more expressive ontology languages: The query answering problem. *Artif. Intell.* 193 (2012), 87–128.

[17] Ashok K. Chandra and Philip M. Merlin. 1977. Optimal Implementation of Conjunctive Queries in Relational Data Bases. In *STOC*. 77–90.

[18] Chandra Chekuri and Anand Rajaraman. 2000. Conjunctive query containment revisited. *Theor. Comput. Sci.* 239, 2 (2000), 211–229.

[19] Bruno Courcelle. 2003. The monadic second-order logic of graphs XIV: uniformly sparse graphs and edge set quantifications. *Theor. Comput. Sci.* 299, 1-3 (2003), 1–36.

[20] Víctor Dalmau, Phokion G. Kolaitis, and Moshe Y. Vardi. 2002. Constraint Satisfaction, Bounded Treewidth, and Finite-Variable Logics. In *CP*. 310–326.

[21] Rodney G. Downey and Michael R. Fellows. 1995. Fixed-Parameter Tractability and Completeness II: On Completeness for W[1]. *Theor. Comput. Sci.* 141, 1&2 (1995), 109–131.

[22] Ronald Fagin, Phokion G. Kolaitis, Renée J. Miller, and Lucian Popa. 2005. Data exchange: semantics and query answering. *Theor. Comput. Sci.* 336, 1 (2005), 89–124.

[23] Georg Gottlob, Marco Manna, and Andreas Pieris. 2014. Polynomial Combined Rewritings for Existential Rules. In *KR*.

[24] Georg Gottlob, Sebastian Rudolph, and Mantas Simkus. 2014. Expressiveness of guarded existential rule languages. In *PODS*. 27–38.

[25] Erich Grädel, Colin Hirsch, and Martin Otto. 2002. Back and forth between guarded and modal logics. *ACM Trans. Comput. Log.* 3, 3 (2002), 418–463.

[26] Martin Grohe. 2007. The complexity of homomorphism and constraint satisfaction problems seen from the other side. *J. ACM* 54, 1 (2007), 1:1–1:24.

[27] Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. 2001. Which Problems Have Strongly Exponential Complexity? *J. Comput. Syst. Sci.* 63, 4 (2001), 512–530.

[28] David S. Johnson and Anthony C. Klug. 1984. Testing Containment of Conjunctive Queries under Functional and Inclusion Dependencies. *J. Comput. Syst. Sci.* 28, 1 (1984), 167–189.

[29] Nicola Leone, Marco Manna, Giorgio Terracina, and Pierfrancesco Veltri. 2019. Fast Query Answering over Existential Rules. *ACM Trans. Comput. Log.* 20, 2 (2019), 12:1–12:48.

[30] David Maier, Alberto O. Mendelzon, and Yehoshua Sagiv. 1979. Testing Implications of Data Dependencies. *ACM Trans. Database Syst.* 4, 4 (1979), 455–469.

[31] Dániel Marx. 2010. Tractable hypergraph properties for constraint satisfaction and conjunctive queries. In *STOC*. 735–744.

[32] W3C OWL Working Group. 2009. *OWL 2 Web Ontology Language: Document Overview*. W3C Recommendation. Available at http://www.w3.org/TR/owl2-overview/.

[33] Christos H. Papadimitriou and Mihalis Yannakakis. 1999. On the Complexity of Database Queries. *J. Comput. Syst. Sci.* 58, 3 (1999), 407–427.

[34] Neil Robertson and Paul D. Seymour. 1986. Graph minors. V. Excluding a planar graph. *J. Comb. Theory, Ser. B* 41, 1 (1986), 92–114.

A PROOF OF PROPOSITION 3.3

The first two items are easy, and have been already discussed in the main body of the paper. Here we focus on item (3). The proof relies on two technical lemmas.

The first such lemma, which is implicit in [15], states that, given an S-database D and an OMQ $Q = (\mathbf{S}, \Sigma, q)$ from $(\mathbb{L}, \mathbb{UCQ})$, $Q(D)$ coincides with the evaluation of q over an initial finite portion C of $\mathsf{chase}(D, \Sigma)$. To formalize this, we need then notion of chase level. Let $s = I_0 \xrightarrow{\sigma_0, (\bar{t}_0, \bar{u}_0)} I_1 \xrightarrow{\sigma_1, (\bar{t}_1, \bar{u}_1)} I_2 \dots$ be a chase sequence for a database D under a set Σ of TGDs, where $\sigma_i = \phi_i(\bar{x}, \bar{y}) \rightarrow \exists \bar{z}\, \psi_i(\bar{x}, \bar{z})$. The *s-level* of an atom $\alpha \in \mathsf{chase}(D, \Sigma)$ is inductively defined as follows: $\mathsf{level}_s(\alpha) = 0$ if $\alpha \in D$, and $\mathsf{level}_s(\alpha) = \ell$ if $\max\{\mathsf{level}_s(\beta) \mid \beta \in \phi_i(\bar{t}_{i-1}, \bar{u}_{i-1})\} = \ell - 1$, where $i > 0$ is such that $\alpha \in I_i$ and $\alpha \notin \bigcup_{0 \leq j < i} I_j$. Let $\mathsf{chase}^{\ell}_s(D, \Sigma)$ be the instance $\{\alpha \in \bigcup_{i \geq 0} I_i \mid \mathsf{level}_s(\alpha) \leq \ell\}$. A chase sequence for D under Σ is *level-wise* if, for $i > 0$, it produces all the atoms of s-level i before generating an atom of s-level $i + 1$.[3]

Lemma A.1. *Consider a database D, a set $\Sigma \in \mathbb{L}$, and a UCQ q. There exists a computable function $g : \mathbb{N} \rightarrow \mathbb{N}$ such that*

$$q(\mathsf{chase}(D, \Sigma)) \;=\; q\left(\mathsf{chase}_s^{g(||\Sigma||+||q||)}(D, \Sigma)\right),$$

where s is a level-wise chase sequence for D under Σ. Furthermore, the instance $\mathsf{chase}_s^{g(||\Sigma||+||q||)}(D, \Sigma)$ can be computed in time $||D|| \cdot f(||Q||)$ for some computable function $f : \mathbb{N} \rightarrow \mathbb{N}$.

Proof. The existence of the computable function g, which is actually an exponential function, has been shown in [15]. It remains to show that $\mathsf{chase}_s^{g(||\Sigma||+||q||)}(D, \Sigma)$ can be computed in time $||D|| \cdot f(||Q||)$ for some computable function $f : \mathbb{N} \rightarrow \mathbb{N}$; recall that s is a level-wise chase sequence for D under Σ. We first provide an upper bound on the size of $\mathsf{chase}_s^{g(||\Sigma||+||q||)}(D, \Sigma)$. Let H_Σ be the maximum number of atoms in the head of a TGD of Σ.

Lemma A.2. *It holds that*

$$\left|\mathsf{chase}_s^{g(||\Sigma||+||q||)}(D, \Sigma)\right| \;\leq\; |D| \cdot (|\Sigma| \cdot H_\Sigma + 1)^{g(||\Sigma||+||q||)}.$$

Proof. We proceed by induction on the s-level $i \geq 0$.

Base case. It is clear that $\mathsf{chase}_s^0(D, \Sigma) = D$, and the claim follows.

Inductive step. Observe that, due to linearity, each TGD of Σ can be triggered by an atom of $\mathsf{chase}_s^{i-1}(D, \Sigma)$ at most once, and generate H_Σ new atoms. Hence,

$$\left|\mathsf{chase}_s^{i}(D, \Sigma)\right| \;\leq\; \left|\mathsf{chase}_s^{i-1}(D, \Sigma)\right| + |\Sigma| \cdot \left|\mathsf{chase}_s^{i-1}(D, \Sigma)\right| \cdot H_\Sigma.$$

By induction hypothesis,

$$\left|\mathsf{chase}_s^{i-1}(D, \Sigma)\right| \;\leq\; |D| \cdot (|\Sigma| \cdot H_\Sigma + 1)^{i-1}.$$

Therefore,

$$\begin{aligned} & \left|\mathsf{chase}_s^{i}(D, \Sigma)\right| \\ \leq\; & |D| \cdot (|\Sigma| \cdot H_\Sigma + 1)^{i-1} + |\Sigma| \cdot |D| \cdot (|\Sigma| \cdot H_\Sigma + 1)^{i-1} \cdot H_\Sigma \\ =\; & |D| \cdot (|\Sigma| \cdot H_\Sigma + 1)^{i-1} \cdot (|\Sigma| \cdot H_\Sigma + 1) \\ =\; & |D| \cdot (|\Sigma| \cdot H_\Sigma + 1)^{i}, \end{aligned}$$

and the claim follows. □

[3] We keep the definition of level-wise chase sequences somewhat informal since it is clear what the underlying intention is.

We can now show that $\mathsf{chase}_s^{g(||\Sigma||+||q||)}(D, \Sigma)$ can be computed in time $||D|| \cdot f(||Q||)$ for some computable function $f : \mathbb{N} \rightarrow \mathbb{N}$. It is easy to see that, for each $i > 0$, $\mathsf{chase}_s^{i}(D, \Sigma)$ can be computed from $\mathsf{chase}_s^{i-1}(D, \Sigma)$ in time (assume that Σ is over the schema $\mathbf{T}$)

$$|\Sigma| \cdot \left|\mathsf{chase}_s^{i-1}(D, \Sigma)\right| \cdot (\mathsf{ar}(\mathbf{T}) + 1) \cdot H_\Sigma \cdot (\mathsf{ar}(\mathbf{T}) + 1).$$

Indeed, for each linear TGD $\sigma \in \Sigma$, we need to scan the instance $\mathsf{chase}_s^{i-1}(D, \Sigma)$, and check whether the atom $\mathsf{body}(\sigma)$ matches with an atom of $\mathsf{chase}_s^{i-1}(D, \Sigma)$, and, if this is the case, add to the instance under construction H_Σ new atoms. By Lemma A.2, for each $i \geq 0$,

$$\begin{aligned} \mathsf{chase}_s^{i}(D, \Sigma) \;&\leq\; \mathsf{chase}_s^{g(||\Sigma||+||q||)}(D, \Sigma) \\ &\leq\; |D| \cdot (|\Sigma| \cdot H_\Sigma + 1)^{g(||\Sigma||+||q||)}. \end{aligned}$$

Thus, $\mathsf{chase}_s^{g(||\Sigma||+||q||)}(D, \Sigma)$ can be computed in time

$$|D| \cdot g(||\Sigma|| + ||q||) \cdot (|\Sigma| \cdot H_\Sigma)^{g(||\Sigma||+||q||)} \cdot |\Sigma| \cdot H_\Sigma \cdot (\mathsf{ar}(\mathbf{T}) + 1)^2.$$

This completes the proof of Lemma A.1. □

The second technical lemma the we need to complete the proof of item (3) of Proposition 3.3 follows:

Lemma A.3. *Let D be an S-database, and $Q = (\mathbf{S}, \Sigma, q) \in (\mathbb{G}, \mathbb{UCQ})$. There exists a database D^*, and a set $\Sigma^* \in \mathbb{L}$ such that*

$$Q(D) \;=\; q(\mathsf{chase}(D^*, \Sigma^*)).$$

Furthermore, D^* can be computed in time $||D||^{O(1)} \cdot f(||Q||)$, and Σ^* in time $g(||Q||)$, for some computable functions $f, g : \mathbb{N} \rightarrow \mathbb{N}$.

Before giving the proof of the above result, let us explain how Lemma A.1 and Lemma A.3 allow us to obtain item (3) of Proposition 3.3. Given an OMQ $Q = (\mathbf{S}, \Sigma, q(\bar{x})) \in (\mathbb{G}, \mathbb{UCQ}_k)$, for $k \geq 1$, an S-database D, and $\bar{c} \in \mathsf{dom}(D)^{|\bar{x}|}$, by Lemmas A.1 and A.3, we simply need to construct a finite instance I in time $||D||^{O(1)} \cdot f(||Q||)$ for some computable function $f : \mathbb{N} \rightarrow \mathbb{N}$, and then check if $\bar{c} \in q(I)$. Since $q \in \mathbb{UCQ}_k$, by Proposition 2.1, the overall procedure takes time $||D||^{O(1)} \cdot g(||Q||)$ for some computable function $g : \mathbb{N} \rightarrow \mathbb{N}$.

A.1 Proof of Lemma A.3

The rest of this section is devoted to the proof of Lemma A.3. Consider an S-database D, and an OMQ $Q = (\mathbf{S}, \Sigma, q) \in (\mathbb{G}, \mathbb{UCQ})$. We proceed to construct a database D^*, and a set $\Sigma^* \in \mathbb{L}$, by adapting a construction from [23], such that $Q(D) = q(\mathsf{chase}(D^*, \Sigma^*))$. Let us first explain the high-level idea underlying D^* and Σ^*.

The High-level Idea. A key notion when reasoning with guarded TGDs is the *type* of an atom α (w.r.t. D and Σ), denoted $\mathsf{type}_{D,\Sigma}(\alpha)$, defined as the set $\{\beta \in \mathsf{chase}(D, \Sigma) \mid \mathsf{dom}(\beta) \subseteq \mathsf{dom}(\alpha)\}$. Roughly speaking, the key property of the type is that the atoms that can be derived during the chase from an atom α used as a guard is determined by its type; see [15] for further details. The main idea underlying the construction of D^* is to encode an atom $\alpha \in D$ and its type as a single atom of the form $[\tau](\cdot)$, where τ is essentially a representation of the "shape" of α and its type. For example, given an atom $R(a, b) \in D$, and assuming that its type is $\{R(a, b), S(b, a), T(a), T(b)\}$, we can encode $R(a, b)$ and its type as the atom $[R(1, 2), \{S(2, 1), T(1), T(2)\}](a, b)$. When it comes to Σ^*, the intention is, for a TGD $\sigma \in \Sigma$, to encode the shape of the type τ of the guard of σ in a predicate $[\tau]$, and then replace σ with a

linear TGD that uses in its body an atom of the form $[\tau](\cdot)$. However, we need an effective way to compute the type of an atom α by completing its known part, which is inherited from the type of the guard atom that generates α, with atoms that mention the new nulls invented in α. This relies on the main property of the type mentioned above. In particular, for each $\alpha \in \mathsf{chase}(D, \Sigma)$ obtained due to the application of the TGD $\sigma = \varphi(\bar{x}, \bar{y}) \rightarrow \exists \bar{z}\, \psi(\bar{x}, \bar{z}) \in \Sigma$ with the witness $(\bar{t}, \bar{u})$, we can construct $\mathsf{type}_{D,\Sigma}(\alpha)$ from $\psi(\bar{t}, \bar{v})$, where $\bar{v}$ is a tuple of new null values, together with the restriction of $\mathsf{type}_{D,\Sigma}(\alpha)$ to the terms $\bar{t}$. This is precisely how we are going to generate new types from existing ones. Before giving the formal construction, we need to introduce some auxiliary terminology.

Auxiliary Terminology. Let $\mathsf{sch}(\Sigma)$ be the set of predicates occurring in Σ. For an atom α, let $\mathsf{base}(\alpha, \Sigma)$ be the set of all atoms that can be formed using terms from $\mathsf{dom}(\alpha)$ and predicates from $\mathsf{sch}(\Sigma)$. A Σ-*type* τ is a pair (α, T), where $\alpha = R(t_1, \ldots, t_n)$ with $R \in \mathsf{sch}(\Sigma)$, $t_1 = 1$ and $t_i \in \{t_1, \ldots, t_{i-1}, t_{i-1} + 1\}$ for each $i \in \{2, \ldots, n\}$, and $T \subseteq \mathsf{base}(\{R(t_1, \ldots, t_n)\}, \Sigma) \setminus \{\alpha\}$. We write $\mathsf{guard}(\tau)$ for the atom α, and $\mathsf{atoms}(\tau)$ for $(\{\alpha\} \cup T)$. The *arity* of τ, denoted $\mathsf{ar}(\tau)$, is the maximum integer occurring in $\mathsf{guard}(\tau)$. Intuitively, τ encodes the shape of a guard atom α and a set of side atoms that are "covered" by α. Consider a tuple $\bar{u} = (u_1, \ldots, u_n)$ that is isomorphic to $\bar{t} = (t_1, \ldots, t_n)$, written $\bar{u} \simeq \bar{t}$, which means that $u_i = u_j$ iff $t_i = t_j$. The *instantiation* of τ with $\bar{u}$, denoted $\tau(\bar{u})$, is the set of atoms obtained from $\mathsf{atoms}(\tau)$ after replacing each t_i with u_i. The *projection* of τ over $P \subseteq \{1, \ldots, \mathsf{ar}(\tau)\}$ is the set of atoms $\Pi_P(\tau) = \{\beta \in \mathsf{atoms}(\tau) \mid \mathsf{dom}(\beta) \subseteq P\}$. The *completion* of an instance I w.r.t. Σ, denoted $\mathsf{complete}(I, \Sigma)$, is defined as the instance $\{R(\bar{t}) \mid \bar{t} \in \mathsf{dom}(I)^{\mathsf{ar}(R)} \text{ and } \{R(\bar{t})\} \rightarrow \mathsf{chase}(I, \Sigma)\}$.

The Formal Construction. The new database is defined as

$$D^* \quad = \quad \left\{ [\tau](\bar{c}) \;\middle|\; \begin{array}{l} R(\bar{c}) \in D \\ \tau = (R(\bar{t}), \cdot) \text{ with } \bar{c} \simeq \bar{t} \\ \tau(\bar{c}) \subseteq \mathsf{complete}(D, \Sigma) \end{array} \right\}.$$

The new set Σ^* of linear TGDs is the union of Σ^*_{tg} and Σ^*_{ex}, where

(1) Σ^*_{tg} is the *type generator*, i.e., is responsible for generating new Σ-types from existing ones.
(2) Σ^*_{ex} is the *expander*, it expands a derived Σ-type τ, i.e., it explicitly constructs the atoms over $\mathsf{sch}(\Sigma)$ encoded by τ.

The type generator is defined as follows. For each $\sigma \in \Sigma$

$$\varphi(\bar{x}, \bar{y}) \;\rightarrow\; \exists z_1 \cdots \exists z_m\, R_1(\bar{u}_1), \ldots, R_n(\bar{u}_n)$$

with $\mathsf{guard}(\sigma) = G(\bar{u})$, and $(\bar{x} \cup \{z_i\}_{i \in [m]})$ the variables occurring in $\mathsf{head}(\sigma)$, and for every Σ-type τ such that there is a homomorphism h from $\varphi(\bar{x}, \bar{y})$ to $\mathsf{atoms}(\tau)$ and $h(G(\bar{u})) = \mathsf{guard}(\tau)$, we add to Σ^*_{tg}

$$[\tau](\bar{u}) \;\rightarrow\; \exists z_1 \cdots \exists z_m\, [\tau_1](\bar{u}_1), \ldots, [\tau_n](\bar{u}_n),$$

where, having the function f with

$$f(t) \;=\; \begin{cases} h(t) & \text{if } t \in \bar{x} \\ \\ \mathsf{ar}(\Sigma) + i & \text{if } t = z_i, \end{cases}$$

as well as the function ρ that renames the integers in atoms in order to appear in increasing order starting from 1 (e.g., $\rho(R(2, 2, 4, 1)) = R(1, 1, 2, 3)$), for each $i \in [n]$, with $\alpha_i = R_i(f(\bar{u}_i))$, the Σ-type τ_i is

$$\big(\rho(\alpha_i), \{\beta \in \mathsf{complete}(I, \Sigma) \mid \mathsf{dom}(\beta) \subseteq \mathsf{dom}(\rho(\alpha_i))\} \setminus \{\rho(\alpha_i)\}\big)$$

with

$$I \;=\; \rho(\{\alpha_i\}_{i \in \{1, \ldots, n\}} \cup \Pi_{\{h(x) \mid x \in \bar{x}\}}(\tau)).$$

The expander it actually constructs, for each Σ-type τ, the guard atom of τ. More precisely, for each Σ-type τ, we add to Σ^*_{ex}

$$[\tau](x_1, \ldots, x_k) \;\rightarrow\; R(x_1, \ldots, x_k),$$

where R is the k-ary predicate of $\mathsf{guard}(\tau)$.

This completes the construction of Σ^*. It is clear that $\Sigma^* \in \mathbb{L}$. It also not difficult to show, by exploiting the properties of the type, that $Q(D) = q(\mathsf{chase}(D^*, \Sigma^*))$, as needed. It remains to show that D^* and Σ^* can be constructed in the claimed times.

An FPT Construction. By construction, Σ^* depends only on Σ, and thus, it is clear that it can be computed in time $g(||Q||)$ for some computable function $g : \mathbb{N} \rightarrow \mathbb{N}$. On the other hand, D^* depends both on D and Σ. It remains to show that D^* can be computed in time $||D||^{O(1)} \cdot f(||Q||)$ for some computable function $f : \mathbb{N} \rightarrow \mathbb{N}$. To this end, we first show the following auxiliary result:

Lemma A.4. *For a database D' and set $\Sigma' \in \mathbb{G} \cap \mathbb{FULL}$, the (finite) instance $\mathsf{chase}(D', \Sigma')$ can be constructed in time $||D'||^{O(1)} \cdot g(||\Sigma'||)$ for some computable function $g : \mathbb{N} \rightarrow \mathbb{N}$.*

Proof. We provide similar analysis as in the proof of Lemma A.1. Let us first establish an upper bound on the size of $\mathsf{chase}(D', \Sigma')$. We assume that Σ' is over the schema $\mathbf{T}$. We can show that:

$$|\mathsf{chase}(D', \Sigma')| \;\leq\; |D'| \cdot |\mathbf{T}| \cdot \mathsf{ar}(\mathbf{T})^{\mathsf{ar}(\mathbf{T})}.$$

Observe that, due to guardedness, two constants $c, d \in \mathsf{dom}(D')$ can occur together in an atom of $\mathsf{chase}(D', \Sigma')$ only if they already occur together in an atom of D'. Thus, for each n-tuple $\bar{c}$ occurring in D' (i.e., there is an atom $R(\bar{c})$ in D'), the chase can produce at most $n^{\mathsf{ar}(\mathbf{T})}$ new tuples of constants from $\bar{c}$, while each new such tuple can be stored in a predicate of $\mathbf{T}$. Since $n \leq \mathsf{ar}(\mathbf{T})$, we get that $\mathsf{chase}(D', \Sigma')$ can have at most $|D'| \cdot |\mathbf{T}| \cdot \mathsf{ar}(\mathbf{T})^{\mathsf{ar}(\mathbf{T})}$ atoms.

We can now show that indeed $\mathsf{chase}(D', \Sigma')$ can be constructed in time $||D||^{O(1)} \cdot g(||Q||)$. Let $B_{\Sigma'}$ (resp., $H_{\Sigma'}$) be the maximum number of atoms in the body (resp., head) of a TGD of Σ'. Observe that, for $i \geq 0$, $\mathsf{chase}^i_s(D', \Sigma')$, where s is a level-wise chase sequence for D' under Σ', can be computed from $\mathsf{chase}^{i-1}_s(D', \Sigma')$ in time

$$|\Sigma'| \cdot \Big(\big|\mathsf{chase}^{i-1}_s(D', \Sigma')\big| \cdot (\mathsf{ar}(\mathbf{T}) + 1) \cdot B_{\Sigma'} \cdot (\mathsf{ar}(\mathbf{T}) + 1) + \big|\mathsf{chase}^{i-1}_s(D', \Sigma')\big| \cdot B_{\Sigma'} \cdot (\mathsf{ar}(\mathbf{T}) + 1)\Big) \cdot H_{\Sigma'} \cdot (\mathsf{ar}(\mathbf{T}) + 1).$$

Indeed, for each guarded TGD $\sigma \in \Sigma'$, we need to scan the instance $\mathsf{chase}^{i-1}_s(D', \Sigma')$, and check whether the atom $\mathsf{guard}(\sigma)$ matches with an atom $\alpha \in \mathsf{chase}^{i-1}_s(D', \Sigma')$. Note that the matching of $\mathsf{guard}(\sigma)$ with α uniquely determines the atoms $\alpha_1, \ldots, \alpha_m$ with which the atoms of $\mathsf{body}(\sigma)$ apart from $\mathsf{guard}(\sigma)$ should match. Then, for each such atom α_i, we need to scan $\mathsf{chase}^{i-1}_s(D', \Sigma')$, and check that indeed it is present. If all the atoms $\alpha_1, \ldots, \alpha_m$ occur in $\mathsf{chase}^{i-1}_s(D', \Sigma')$, then we add $H_{\Sigma'}$ new atoms.

As shown above, for each $i \geq 0$,

$$\big|\mathsf{chase}^i_s(D', \Sigma')\big| \;\leq\; |\mathsf{chase}(D', \Sigma')| \;\leq\; |D'| \cdot |\mathbf{T}| \cdot \mathsf{ar}(\mathbf{T})^{\mathsf{ar}(\mathbf{T})}.$$

Since $\mathsf{chase}(D', \Sigma')$ can have at most $n = |D'| \cdot |\mathbf{T}| \cdot \mathsf{ar}(\mathbf{T})^{\mathsf{ar}(\mathbf{T})}$ s-levels, it can be computed in time

$$n \cdot |\Sigma'| \cdot (n \cdot (\mathsf{ar}(\mathbf{T}) + 1) + B_{\Sigma'} \cdot (\mathsf{ar}(\mathbf{T}) + 1) + n \cdot B_{\Sigma'} \cdot (\mathsf{ar}(\mathbf{T}) + 1)) \cdot H_{\Sigma'} \cdot (\mathsf{ar}(\mathbf{T}) + 1).$$

Summing up, $\mathsf{chase}(D', \Sigma')$ can be computed in time

$$||D'||^2 \cdot g(||\Sigma'||)$$

for some computable function $g : \mathbb{N} \to \mathbb{N}$, and the claim follows. □

In view of Lemma A.4, to show that D^* can be computed in time $||D||^{O(1)} \cdot f(||Q||)$, it suffices to show that we can construct in time $g(||\Sigma||)$ a set $\hat{\Sigma} \in \mathbb{G} \cap \mathbb{FULL}$ such that $D^* = \mathsf{chase}(D, \hat{\Sigma})_{|\mathbf{T}}$, where $\mathbf{T}$ consists of all the predicates of the form $[\tau]$, where τ is a Σ-type, occurring in $\mathsf{chase}(D, \hat{\Sigma})$. The set $\hat{\Sigma}$ will simply:

(1) complete the database D with all the atoms of $\mathsf{chase}(D, \Sigma)$ that mention only constants from $\mathsf{dom}(D)$, i.e., add to D the set of atoms $\mathsf{chase}(D, \Sigma) \cap \mathsf{base}(D, \Sigma)$, and
(2) generate all the atoms of D^*.

For achieving task (1) above, we exploit a result from [24], which states the following: given a set $\Sigma' \in \mathbb{G}$, we can construct a set $\xi(\Sigma') \in \mathbb{G} \cap \mathbb{FULL}$ over $\mathsf{sch}(\Sigma')$ such that, for every database D', $\mathsf{chase}(D', \xi(\Sigma')) = \mathsf{chase}(D', \Sigma') \cap \mathsf{base}(D', \Sigma')$. Thus, task (1) can be done via the set $\xi(\Sigma) \in \mathbb{G}$ over $\mathsf{sch}(\Sigma)$.

Task (2) can be achieved via the set of TGDs $\Sigma_{types} \in \mathbb{G} \cap \mathbb{FULL}$ defined as follows. For each Σ-type τ, we have a TGD

$$\tau(x_1, \ldots, x_k) \rightarrow [\tau]\left(x_{f_\tau(1)}, \ldots, x_{f_\tau(\ell)}\right),$$

where $k = \mathsf{ar}(\tau)$, ℓ is the arity of the predicate of $\mathsf{guard}(\tau)$, and, assuming that $(i_1, \ldots, i_\ell)$ is the tuple of $\mathsf{guard}(\tau)$, $f_\tau(i) = i_j$, for each $j \in [\ell]$. By abuse of notation, we use $\tau(x_1, \ldots, x_k)$ to denote the conjunction of atoms in the instantiation of τ with $(x_1, \ldots, x_k)$.

The set $\hat{\Sigma}$ is defined as $\xi(\Sigma) \cup \Sigma_{types}$, which can be clearly constructed in time $g(||\Sigma||)$ since it depends only on Σ.

B PROOF OF THEOREM 5.1

Given a CQ q, we write $\mathsf{var}(q)$ for the set of variables in q. We may also treat, as usual, a CQ q as the set of atoms in q. The proofs in this section rely on notions and results introduced in the proof of Proposition 5.2, which we advise the reader to read first.

The 2ExpTime-hardness is inherited from [7], where the same problem for OMQs based on DLs has been studied. It remains to establish the 2ExpTime upper bound. Let $Q = (\mathbf{S}, \Sigma, q)$. Proposition 5.2 provides a procedure for deciding (uniform) $\mathbb{UCQ}_k$-equivalence:

(1) compute the $\mathbb{UCQ}_k$-approximation $Q_k^a = (\mathbf{S}, \Sigma, q_k^a)$ of Q;
(2) accept if $Q \equiv Q_k^a$; otherwise, reject.

This yields decidability. It also shows that if Q is $\mathbb{UCQ}_k$-equivalent, then an OMQ Q' from $(\mathbb{G}, \mathbb{UCQ}_k)$ such that $Q \equiv Q'$ can be constructed in double exponential time. Actually, to find Q' we can simply construct the OMQ Q_k^a.

Lemma B.1. *Q_k^a can be constructed in double exponential time.*

Proof. For each disjunct p of q, we first need to compute its specializations (p', V). It is easy to see that there are single exponentially many specialization that can be found in single exponential time. For each specialization $s = (p', V)$, we then have to find all Σ-groundings $g_s(\bar{x})$ of s, and keep the ones of treewidth at most k. Note that we can form double exponentially many guarded CQs, each of exponential size. Thus, there are double exponentially many candidates for being a Σ-grounding of s, each for exponential size, that can be found in double exponential time. Now, in the check whether such a candidate g is indeed a Σ-grounding of s, the only non-trivial part is that we have to check for each maximally $[V]$-connected component p_i of $p'[V]$ whether $p_i \rightarrow \mathsf{chase}(g_i, \Sigma)$, where g_i is a guarded subquery of g of exponential size, via a homomorphism that is the identity on $\mathsf{var}(p_i) \cap V = \{y_1, \ldots, y_m\}$. This boils down to the problem whether $(y_1, \ldots, y_m) \in p_i'(\mathsf{chase}(g_i, \Sigma))$, where $p_i'(y_1, \ldots, y_m)$ consists of the atoms of p_i. By item (2) of Proposition 3.2, this can be checked in 2ExpTime. Actually, this is not immediate from Proposition 3.2 since g_i might be of exponential size. We additionally need to say that the double exponential bound obtained from Proposition 3.2 is only polynomial in the size of the database; implicit in [14]. Finally, checking whether a Σ-grounding g_s of s has treewidth at most k is feasible in double exponential time since g_s is, in general, of exponential size. □

Containment between OMQs from $(\mathbb{G}, \mathbb{UCQ})$ can be decided in 2ExpTime [6]. Since, however, q_k^a may consists of double exponentially many CQs, it is not clear how to implement step (2) above in 2ExpTime and how to obtain the 2ExpTime upper bound in Theorem 5.1 by a direct implementation of the above procedure. We solve this problem in two steps. First, we replace Q_k^a with an OMQ $Q_k' = (\mathbf{S}, \Sigma', q_k')$ that is equivalent to Q_k^a, and such that q_k' contains only single exponentially many CQs, each of polynomial size. And second, we modify the containment check from [6] in a mild way.

B.1 Replacing Q_k^a with Q_k'

The OMQ $Q_k' = (\mathbf{S}, \Sigma', q_k')$ is defined as follows. We introduce a fresh unary relation symbol A (that is not in $\mathbf{S}$) and obtain Σ' from Σ by extending every TGD head with the atom $A(x)$ whenever x is an existentially quantified variable in that TGD head. Then q_k' is the UCQ that includes p_c for all specializations $s = (p_c, V)$ of a CQ p in q such that there exists a Σ-grounding of s that is of treewidth at most k, extended by adding $A(x)$ for every variable $x \in \mathsf{var}(p_c) \setminus V$.

It is clear that q_k' has only single exponentially many disjuncts, each of polynomial size. Reusing the arguments from the proof of Lemma B.1, it is also clear that Q_k' can be constructed in double exponential time. We further make the following important observation, which rests on the assumption that $k \geq \mathsf{ar}(\mathbf{T}) - 1$.

Lemma B.2. *Let $s = (p_c, V)$ be a specialization of a CQ in q. If there is a Σ-grounding of s that has treewidth at most k, then all Σ-groundings of s have treewidth at most k.*

Proof. Let $p_1, ..., p_n$ be the maximally $[V]$-connected components of $p_c[V]$, let $g_s(\bar{x}) = \exists \bar{z}\,(g_0 \wedge g_1 \wedge \cdots \wedge g_n)$ be a Σ-grounding of s, and let $\delta = (T_\delta, \chi)$, where $T_\delta = (V_\delta, E_\delta)$, be a tree decomposition of $G^{g_s}_{|\bar{y}}$ of width k, $\bar{y}$ the existentially quantified variables of $g_s(\bar{x})$. For $i \in [n]$, there must be a $v_i \in V_\delta$ with $\mathsf{var}(p_i) \cap V \subseteq \chi(v_i)$. Now, let $g_s'(\bar{x}) = \exists \bar{z}\,(g_0 \wedge g_1' \wedge \cdots \wedge g_n')$ be another Σ-grounding of s. Let $\delta' = (T_{\delta'}, \chi')$ be obtained from δ by dropping all variables that occur in g_s but not in g_s' from every bag and adding, for $i \in [n]$, a fresh successor u_i to v_i and setting $\chi'(u_i) = \mathsf{var}(g_i')$. It can be

verified that δ' is a tree decomposition of $G^{g'_s}_{|\bar{y}'}$, $\bar{y}'$ the existentially quantified variables of g'_s. In particular, Condition (2) of tree decompositions is satisfied for every edge induced by an atom in some g'_i due to the presence of u_i. Since g'_i is guarded, the number of variables in $\mathsf{var}(g'_i)$ is bounded by $\mathsf{ar}(\mathbf{T})$. Since $k \geq \mathsf{ar}(\mathbf{T}) - 1$, the width of δ' is bounded by k, and the claim follows. □

We now proceed to show that Q^a_k and Q'_k are indeed equivalent.

Lemma B.3. $Q^a_k \equiv Q'_k$.

Proof. First assume that $\bar{c} \in Q'_k(D)$ with D an $\mathbf{S}$-database. Then, there is a homomorphism h from a CQ $p(\bar{x})$ in q'_k to $\mathsf{chase}(D, \Sigma')$ with $h(\bar{x}) = \bar{c}$. Assume that p was constructed for the specialization $s = (p_c, V)$, that is, p is the extension of p_c with $A(x)$ for all $x \in \mathsf{var}(p_c) \setminus V$. By construction of Σ', h thus maps all variables in $\mathsf{var}(p) \setminus V$ to constants that the chase has introduced to satisfy existential quantifiers in TGD heads. Let $p_1, \ldots, p_n$ be the maximally $[V]$-connected components of $p_c[V]$. It follows from the facts that p_i is $[V]$-connected, and that h maps all variables in $\mathsf{var}(p) \setminus V$ to existential constants that we find a fact $\alpha_i \in D$ such that the restriction of h to the variables in p_i is a homomorphism to $\mathsf{chase}(\mathsf{type}_{D,\Sigma'}(\alpha_i), \Sigma')$, for $i \in [n]$. Let the Σ-grounding g_s of s by obtained by using as g_i the guarded full CQ that is obtained by viewing $\mathsf{type}_{D,\Sigma'}(\alpha_i)$ as a full CQ. Clearly, the restriction of h to the variables in V can be extended to a homomorphism h' from g_s to $\mathsf{chase}(D, \Sigma')$ by mapping the variables from the CQs g_i that are not in V to constants from α_i in the expected way. Since g_s does not contain the relation A, h' maps g_s to $\mathsf{chase}(D, \Sigma)$. It remains to argue that g_s is a CQ in q^a_k. Since p is in q'_k, there is a Σ-grounding of s that is of treewidth at most k. By Lemma B.2, this implies that g_s is of treewidth at most k and thus g_s is in q^a_k.

Conversely, assume that $\bar{c} \in Q^a_k(D)$ with D an $\mathbf{S}$-database. Then, there is a homomorphism h from a CQ $g_s(\bar{x})$ in q^a_k to $\mathsf{chase}(D, \Sigma)$ with $h(\bar{x}) = \bar{c}$. Let g_s be a Σ-grounding of the specialization $s = (p_c, V)$ of a CQ p in q. Then g_s is of treewidth at most k. Consequently, s gives rise to a corresponding CQ p' in q'_k, that is, p' is the extension of p_c with $A(x)$ for all $x \in \mathsf{var}(p_c) \setminus V$. Clearly, the restriction of $h_{|_V}$ to the variables in V is a homomorphism from $p'_{|_V}$ to $\mathsf{chase}(D, \Sigma)$. Let $p_1, \ldots, p_n$ be the maximally $[V]$-connected components of $p_c[V]$. By definition of Σ-groundings, $p_i \to \mathsf{chase}(g_i, \Sigma)$ via a homomorphism that is the identity on $\mathsf{var}(p_i) \cap V$, for $i \in [n]$. We can combine all these homomorphisms with $h_{|_V}$ to obtain a homomorphism h' from p_c to $\mathsf{chase}(D, \Sigma)$ that maps all variables outside of V to constants that have been introduced by the chase to satisfy existential quantifiers in TGD heads. By construction of p' and Σ', h' is also a homomorphism from p' to $\mathsf{chase}(D, \Sigma')$. □

B.2 Checking Equivalence

We now proceed to analyze the complexity of checking whether $Q \equiv Q'_k$. By item (1) of Lemma C.7 and Lemma B.3, $Q'_k \subseteq Q$. The non-trivial task is to check whether $Q \subseteq Q'_k$. We know from [6] that checking containment among OMQs from $(\mathbb{G}, \mathbb{UCQ})$ is in 2ExpTime. In fact, [6] first provides a polynomial time reduction from containment among OMQs from $(\mathbb{G}, \mathbb{UCQ})$, denoted $\mathsf{Cont}(\mathbb{G}, \mathbb{UCQ})$, to containment among OMQs from $(\mathbb{G}, \mathbb{CQ})$, denoted $\mathsf{Cont}(\mathbb{G}, \mathbb{CQ})$, and then devises an automata-based procedure for $\mathsf{Cont}(\mathbb{G}, \mathbb{CQ})$ that runs in double exponential time in the combined size of the OMQs. However, Q'_k might be of exponential size, which implies that if we first apply the reduction from $\mathsf{Cont}(\mathbb{G}, \mathbb{UCQ})$ to $\mathsf{Cont}(\mathbb{G}, \mathbb{CQ})$, we get an OMQ that has a CQ of exponential size, and then, by using the decision procedure for $\mathsf{Cont}(\mathbb{G}, \mathbb{CQ})$ from [6] as a black box, we only get a 3ExpTime upper bound. Nevertheless, it turns out that we can reuse the automata constructions from [6] for $\mathsf{Cont}(\mathbb{G}, \mathbb{CQ})$ in order to devise a procedure that directly operates on OMQs from $(\mathbb{G}, \mathbb{UCQ})$ and decides containment in time double exponential in the size of the ontology, the schema, and the maximum size of the CQs in the UCQs in the OMQs, but only single exponential in the number of disjuncts in those UCQs. Having such a procedure in place, it is then clear that checking whether $Q \subseteq Q'_k$ is in 2ExpTime since, due to Lemma B.1, and since although the UCQ in Q'_k has exponentially many disjuncts, each one is of polynomial size. For the sake of completeness, we recall the key ingredients underlying the automata-based procedure from [6], and then explain how we get the automata-based procedure that directly operates on OMQs from $(\mathbb{G}, \mathbb{UCQ})$ that leads to the desired upper bound.

The Automata-based Procedure for $\mathsf{Cont}(\mathbb{G}, \mathbb{UCQ})$

Let us first recall that we can focus, w.l.o.g., on Boolean OMQs since there is an easy reduction, originally proposed in [10], of containment for non-Boolean OMQs to containment for Boolean OMQs. In what follows, we write $\mathbb{UCQ}$ and $\mathbb{CQ}$ meaning the classes of Boolean UCQs and Boolean CQs, respectively.

We know from [6] that, given two OMQs Q_1 and Q_2 from $(\mathbb{G}, \mathbb{CQ})$, if $Q_1 \not\subseteq Q_2$, then this is witnessed via a "nearly acyclic" database. To formalize this we need the notion of guarded tree decomposition. A tree decomposition $\delta = (T_\delta, \chi)$, where $T_\delta = (V_\delta, E_\delta)$, of G^D for some database D is called $[V]$-*guarded*, where $V \subseteq V_\delta$, if for every node $v \in V_\delta \setminus V$, there exists a fact $R(c_1, \ldots, c_n) \in D$ such that $\chi(v) \subseteq \{c_1, \ldots, c_n\}$, i.e., $\chi(v)$ is guarded in D. We write $\mathsf{root}(\delta)$ for the root node of δ, and $D_\delta(v)$, for $v \in V_\delta$, for the subset of D induced by $\chi(v)$. An $\mathbf{S}$-database is called a C-*tree*, where $C \subseteq D$, if there exists a tree decomposition δ of G^D such that: (i) $D_\delta(\mathsf{root}(\delta)) = C$, and (ii) δ is $[\{\mathsf{root}(\delta)\}]$-guarded. Roughly, if a database D is a C-tree, then C can be viewed as the cyclic part of D, while the rest of D is acyclic. In [6], it has been shown that for containment purposes we can focus on databased that are C-trees with the size of C bounded and depending only on the left-hand side OMQ. Note, however, that the result of [6] talks only about OMQs from $(\mathbb{G}, \mathbb{CQ})$. A careful inspection of the proof given in [6] reveals that the following statement for OMQs from $(\mathbb{G}, \mathbb{UCQ})$ holds:

Lemma B.4. *Let $Q_1 = (\mathbf{S}, \Sigma_1, q_1)$ and $Q_2 = (\mathbf{S}, \Sigma_2, q_2)$ be OMQs from $(\mathbb{G}, \mathbb{UCQ})$. The following are equivalent:*

1. *$Q_1 \not\subseteq Q_2$.*
2. *There exists a C-tree $\mathbf{S}$-database D with $||\mathsf{dom}(C)|| \leq \mathsf{ar}(\mathbf{S} \cup \mathsf{sch}(\Sigma_1)) \cdot \max_{p \in q_1}\{||p||\}$ such that $Q_1(D) \not\subseteq Q_2(D)$.*

The goal is to devise a two-way alternating parity tree automaton (2ATA) that checks for the second item of Lemma B.4. To this end, we need a convenient encoding of a C-tree database as a tree over a finite alphabet that can be accepted by a 2ATA. As shown in [6], this can be done by exploiting standard encodings of bounded treewidth

databases. In a nutshell, a C-tree S-database D can be encoded as a finite $\Gamma_{\mathbf{S},\ell}$-labeled tree, with $||\mathsf{dom}(C)|| \leq \ell$ and $\Gamma_{\mathbf{S},\ell}$ being an alphabet of double exponential size in $\mathsf{ar}(\mathbf{S})$, and exponential in $||\mathbf{S}||$ and ℓ; the actual encoding is not important for our discussion. Of course, it is possible that a $\Gamma_{\mathbf{S},\ell}$-labeled tree does not encode a C-tree database. Thus, some additional consistency properties are needed, which are also not important for our discussion. A $\Gamma_{\mathbf{S},\ell}$-labeled tree is *consistent* if it satisfies those syntactic properties, and it can be shown that a consistent $\Gamma_{\mathbf{S},\ell}$-labeled tree L can be decoded into an S-database, denoted $[L]$, that is a C-tree with $||\mathsf{dom}(C)|| \leq \ell$. Therefore, by Lemma B.4, we get the following:

Lemma B.5. *Let $Q_1 = (\mathbf{S}, \Sigma_1, q_1)$ and $Q_2 = (\mathbf{S}, \Sigma_2, q_2)$ be OMQs from $(\mathbb{G}, \mathbb{UCQ})$. The following are equivalent:*

(1) $Q_1 \not\subseteq Q_2$.
(2) There exists a consistent $\Gamma_{\mathbf{S},\ell}$-labeled tree L, where $\ell = \mathsf{ar}(\mathbf{S} \cup \mathsf{sch}(\Sigma_1)) \cdot \max_{p \in q_1}\{||p||\}$, such that $Q_1([L]) \not\subseteq Q_2([L])$.

It should be clear now that the goal is to devise a 2ATA A over finite trees such that its language, denoted $L(A)$, is the set of $\Gamma_{\mathbf{S},\ell}$-labeled tree L, where $\ell \leq \mathsf{ar}(\mathbf{S} \cup \mathsf{sch}(\Sigma_1)) \cdot \max_{p \in q_1}\{||p||\}$, such that $Q_1([L]) \not\subseteq Q_2([L])$. Having such an automaton in place, checking whether $Q_1 \subseteq Q_2$ boils down to checking whether $L(A)$ is empty, which can be done in exponential time in the number of states of A, and in polynomial time in the size of the input alphabet. Thus, to obtain the desired 2ExpTime upper bound, we need to construct A in double exponential time, while the number of its states should be at most exponential. To construct the desired 2ATA we are going to exploit the automata constructions from [6].

First, we need a way to check consistency of labeled trees, which, as shown in [6], can be done via an automaton.

Lemma B.6. *Consider a schema $\mathbf{S}$, and an integer $\ell > 0$. There is a 2ATA $C_{\mathbf{S},\ell}$ that accepts a $\Gamma_{\mathbf{S},\ell}$-labeled tree L iff L is consistent. The number of states of $C_{\mathbf{S},\ell}$ is logarithmic in the size of $\Gamma_{\mathbf{S},\ell}$. Furthermore, $C_{\mathbf{S},\ell}$ can be constructed in polynomial time in the size of $\Gamma_{\mathbf{S},\ell}$.*

We also know from [6] that, given an OMQ Q from $(\mathbb{G}, \mathbb{CQ})$, we can devise an automaton that accepts labeled trees which correspond to databases that make Q true.

Lemma B.7. *Let $Q = (\mathbf{S}, \Sigma, q) \in (\mathbb{G}, \mathbb{CQ})$. There is a 2ATA $A_{Q,\ell}$, where $\ell > 0$, that accepts a consistent $\Gamma_{\mathbf{S},\ell}$-labeled tree L iff $[L] \models Q$. $A_{Q,\ell}$ has exponentially many states in $||Q||$ and ℓ, and it can be constructed in double exponential time in $||Q||$ and ℓ.*

We are now ready to devise the desired 2ATA. To this end, we are going to exploit Lemmas B.5 to B.7, and some well-known facts about 2ATAs. In particular, languages accepted by 2ATAs are closed under union and complementation. Given two 2ATAs A_1 and A_2, we write $A_1 \cup A_2$ or a 2ATA, which can be constructed in polynomial time, that accepts the language $L(A_1) \cup L(A_2)$. Moreover, we write $\overline{A}$ for the 2ATA, which can also be constructed in polynomial time, that accepts the complement of $L(A)$. Importantly, the number of states of $A_1 \cup A_2$ is the sum of the numbers of states of A_1 and A_2 plus one. Given $Q_1 = (\mathbf{S}, \Sigma_1, q_1)$ and $Q_2 = (\mathbf{S}, \Sigma_2, q_2)$ from $(\mathbb{G}, \mathbb{UCQ})$, we write $Q_{1,2}^{\max}$ for the largest OMQ among $\bigcup_{p \in q_1}(\mathbf{S}, \Sigma_1, p) \cup \bigcup_{p \in q_2}(\mathbf{S}, \Sigma_2, p)$.

Lemma B.8. *Let $Q_1 = (\mathbf{S}, \Sigma_1, q_1)$ and $Q_2 = (\mathbf{S}, \Sigma_2, q_2)$ be OMQs from $(\mathbb{G}, \mathbb{UCQ})$. There is a 2ATA A such that*

$$Q_1 \subseteq Q_2 \iff L(A) = \emptyset.$$

A has exponentially many states in $||Q_{1,2}^{\max}||$ and ℓ, and polynomially many states in $||q_1|| + ||q_2||$. It can be constructed in double exponential time in $||Q_{1,2}^{\max}||$ and ℓ, and in polynomial time in $||q_1|| + ||q_2||$.

Proof. Let $\ell = \mathsf{ar}(\mathbf{S} \cup \mathsf{sch}(\Sigma_1)) \cdot \max_{p \in q_1}\{||p||\}$. The 2ATA A is

$$\left(C_{\mathbf{S},\ell} \cap \left(\bigcup_{p \in q_1} A_{(\mathbf{S},\Sigma_1,p),\ell}\right)\right) \cap \left(\overline{\bigcup_{p \in q_2} A_{(\mathbf{S},\Sigma_2,p),\ell}}\right).$$

Using Lemmas B.5, B.6, and B.7, the fact that union and complementation of 2ATA are feasible in polynomial time, and the stated bound on the number of states in automata for union, it is easy to verify that indeed A is the desired 2ATA. □

It is now easy to see, having Lemma B.8 in place, that indeed checking whether $Q \subseteq Q'_k$ is feasible in double exponential time. This actually follows from the fact that the UCQ of Q'_k has exponentially many disjuncts, each of polynomial size, and the fact that emptiness for 2ATA can be decided in exponential time in the number of states of the automaton.

C PROOF OF PROPOSITION 5.2

$\mathbb{UCQ}_k$-approximations for OMQs from $(\mathbb{G}, \mathbb{UCQ})$ rely on the notion of specialization of a CQ. We first introduce CQ specializations, and give a lemma that illustrates their usefulness. We then define the notion of grounding of a CQ specialization, and establish a technical lemma that will allow us to define $\mathbb{UCQ}_k$-approximations. We then establish our main technical result about $\mathbb{UCQ}_k$-approximations, which in turn allows us to complete the proof of Proposition 5.2. In what follows, given a CQ q, we write $\mathsf{var}(q)$ for the set of variables in q. We may also treat, as usual, a CQ q as the set of atoms in q.

C.1 CQ Specializations

Recall that a contraction of a CQ $q(\bar{x})$ is a CQ $p(\bar{x})$ that can be obtained from q by identifying variables. When an answer variable x is identified with a non-answer variable y, the resulting variable is x; the identification of two answer variables is not allowed. The formal definition of CQ specialization follows:

Definition C.1. Consider a CQ $q(\bar{x})$. A *specialization* of q is a pair $s = (p, V)$, where p is a contraction of q, and $\bar{x} \subseteq V \subseteq \mathsf{var}(p)$. ■

Intuitively speaking, a specialization of a CQ q describes a way how q can be homomorphically mapped to the chase of a database D under a set Σ of TGDs. First, instead of mapping q to $\mathsf{chase}(D, \Sigma)$, we injectively map p to $\mathsf{chase}(D, \Sigma)$, while the set of variables V collects all the variables of p that are mapped to constants of $\mathsf{dom}(D)$. The next result, which relies on the properties of the type of an atom (see the proof of Lemma A.1 in Appendix **??**) illustrates the usefulness of CQ specializations. A CQ q is $[V]$-*connected*, for $V \subseteq \mathsf{var}(q)$, if $G^q_{|\mathsf{var}(q) \setminus V}$, that is, the subgraph of G^q induced by $\mathsf{var}(q) \setminus V$, is connected. We also write $q[V]$ for the subquery of q obtained after dropping the atoms of $q_{|V}$, i.e., the atoms that contain only variables of V. Given a database D, and a set Σ of TGDs, we write $\mathsf{chase}_{\downarrow}(D, \Sigma)$ for the ground part of $\mathsf{chase}(D, \Sigma)$, that is, the set consisting of all atoms in $\mathsf{chase}(D, \Sigma)$ with only constants from $\mathsf{dom}(D)$.

Lemma C.2. *Consider a database D, a set $\Sigma \in \mathbb{G}$, a CQ $q(\bar{x})$, and a tuple $\bar{c} \in \mathsf{dom}(D)^{|\bar{x}|}$. The following are equivalent:*

(1) $\bar{c} \in q(\mathsf{chase}(D, \Sigma))$.

(2) There exists a specialization $s = (p, V)$ *of* q *such that* $p \to \mathsf{chase}(D, \Sigma)$ *via an injective homomorphism* μ *with (i)* $\mu(\bar{x}) = \bar{c}$*, (ii)* $\mu(p_{|V}) \subseteq \mathsf{chase}_{\downarrow}(D, \Sigma)$*, and (iii) for every maximally* $[V]$*-connected component* p' *of* $p[V]$*, there exists an atom* $\alpha \in D$ *such that* $\mathsf{dom}(\alpha) \supseteq \{\mu(y) \mid y \in \mathsf{var}(p') \cap V\}$ *and* $\mu(p') \subseteq \mathsf{chase}(\mathsf{type}_{D,\Sigma}(\alpha), \Sigma)$.

C.2 Grounding Specializations

We now define the notion of grounding of a specialization (p, V) of a CQ q. The intention is to replace each maximally $[V]$-connected component p' of $p[V]$ with a guarded set of atoms, which is ought to be mapped to the ground part of the chase, that entails p' under the given set of TGDs. A CQ q is *guarded* if it has an atom, denoted $\mathsf{guard}(q)$, that contains all the variables of $\mathsf{var}(q)$, and *full* if all the variables of $\mathsf{var}(q)$ are answer variables.

Definition C.3. Consider a set Σ of TGDs over $\mathbf{T}$, and a CQ $q(\bar{x})$ over $\mathbf{T}$. Let $s = (p, V)$ be a specialization of q with $p_1, \ldots, p_n$ being the maximally $[V]$-connected components of $p[V]$. A Σ-*grounding* of s is a CQ $g_s(\bar{x})$ over $\mathbf{T}$ of the form

$$\exists \bar{z}\ (g_0 \wedge g_1 \wedge \cdots \wedge g_n)$$

where

- g_0 is the full CQ consisting of the atoms of $p_{|V}$,
- for each $i \in [n]$, g_i is a guarded full CQ such that:
  - $\mathsf{var}(g_i) \subseteq (\mathsf{var}(p_i) \cap V) \cup \left\{y_1^i, \ldots, y^i_{\mathsf{ar}(\mathbf{T})-m}\right\} \subset \mathbf{V}$,
  - $\mathsf{var}(p_i) \cap V \subseteq \mathsf{var}(\mathsf{guard}(g_i))$, and
  - $p_i \to \mathsf{chase}(g_i, \Sigma)$ via a homomorphism that is the identity on $\mathsf{var}(p_i) \cap V$, and
- $\bar{z} = \left(\bigcup_{0 \le i \le n} \mathsf{var}(g_i)\right) \setminus \bar{x}$. ∎

Before giving the main lemma concerning groundings of CQ specializations, let us establish a useful auxiliary claim.

Claim C.4. *Let* Σ *be a set of TGDs over* $\mathbf{T}$*, and* $q(\bar{x})$ *a CQ over* $\mathbf{T}$*. Let* $g_s(\bar{x})$ *be a* Σ*-grounding of a specialization* $s = (p, V)$ *of* q*, and assume that* $g_s \to I$*, for an instance* I*, via a homomorphism* μ*. Then,* $p \to \mathsf{chase}(I, \Sigma)$ *via a homomorphism* ξ *with* $\mu(\bar{x}) = \xi(\bar{x})$.

Proof. Let $p_1, \ldots, p_n$ be the maximally $[V]$-connected components of $p[V]$. Assume that $g_s(\bar{x})$ is of the form

$$\exists \bar{z}\,(g_0 \wedge g_1 \wedge \cdots \wedge g_n),$$

where g_0 is the full CQ consisting of the atoms of $p_{|V}$, and for each $i \in [n]$, g_i is a guarded full CQ with the properties given in Definition C.3. Clearly, for each $i \in [n]$, there exists a mapping λ_i that is the identity on $\mathsf{var}(p_i) \cap V$ that maps p_i to $\mathsf{chase}(g_i, \Sigma)$. Since the sets of atoms $p_{|V}, p_1, \ldots, p_n$ share only variables of V, we conclude that $\lambda = \bigcup_{i \in [m]} \lambda_i$ is a well-defined mapping that maps p to $\mathsf{chase}(g_s, \Sigma)$ with $\lambda(\bar{x}) = \bar{x}$. Since, by hypothesis, $g_s \to I$ via some μ, it is not difficult to show that $\mathsf{chase}(g_s, \Sigma)$ maps to $\mathsf{chase}(I, \Sigma)$ via μ' that extends μ, and thus, $\mu(\bar{x}) = \mu'(\bar{x})$. Therefore, $\xi = \mu' \circ \lambda$ maps p to $\mathsf{chase}(I, \Sigma)$ with $\mu(\bar{x}) = \xi(\bar{x})$, as needed. □

We are now ready to show the main lemma about groundings of CQ specializations.

Lemma C.5. *Consider a database* D*, a set* $\Sigma \in \mathbb{G}$ *over a schema* $\mathbf{T}$*, a CQ* $q(\bar{x})$ *over* $\mathbf{T}$*, and* $\bar{c} \in \mathsf{dom}(D)^{|\bar{x}|}$*. The following are equivalent:*

(1) $\bar{c} \in q(\mathsf{chase}(D, \Sigma))$.

(2) There exists a specialization s *of* q *such that a* Σ*-grounding* $g_s(\bar{x})$ *of* s *maps to* $\mathsf{chase}_{\downarrow}(D, \Sigma)$ *via an injective mapping* μ *with* $\mu(\bar{x}) = \bar{c}$.

Proof. (1) ⇒ (2). By Lemma C.2, there is a specialization $s = (p, V)$ of $q(\bar{x})$, with $p_1, \ldots, p_n$ being the maximally $[V]$-connected components of $p[V]$, such that $p \to \mathsf{chase}(D, \Sigma)$ via an injective mapping λ with (i) $\lambda(\bar{x}) = \bar{c}$, (ii) $\lambda(p_{|V}) \subseteq \mathsf{chase}_{\downarrow}(D, \Sigma)$, and (iii) for every maximally $[V]$-connected component p' of $p[V]$, there exists an atom $\alpha \in D$ such that $\mathsf{dom}(\alpha) \supseteq \{\lambda(y) \mid y \in \mathsf{var}(p') \cap V\}$ and $\lambda(p') \subseteq \mathsf{chase}(\mathsf{type}_{D,\Sigma}(\alpha), \Sigma)$. Consider the CQ $g_s(\bar{x})$ of the form

$$\exists \bar{z}\ (g_0 \wedge g_1 \wedge \cdots \wedge g_n),$$

where g_0 is the full CQ consisting of the atoms of $p_{|V}$, and, for each $i \in [n]$, g_i is the guarded full CQ obtained from $\mathsf{type}_{D,\Sigma}(\alpha_i)$ by converting $\lambda(y)$, where $y \in \mathsf{var}(p_i) \cap V$, into the variable y, and each constant $c^i_j \in \{c^i_1, \ldots, c^i_{\ell_i}\} = \mathsf{dom}(\mathsf{type}_{D,\Sigma}(\alpha_i)) \setminus \{\lambda(y) \mid y \in \mathsf{var}(p_i) \cap V\}$ into the variable y^i_j. Finally, let $\bar{z} = \left(\bigcup_{0 \le i \le n} \mathsf{var}(g_i)\right) \setminus \bar{x}$. Clearly, $g_s(\bar{x})$ is a Σ-grounding of s. Moreover, the mapping $\mu = \lambda \cup \{y^i_j \mapsto c^i_j\}_{i \in [n], j \in [\ell_i]}$ maps injectively g_s to $\mathsf{chase}_{\downarrow}(D, \Sigma)$, since $\mathsf{type}_{D,\Sigma}(\alpha_i) \subseteq \mathsf{chase}_{\downarrow}(D, \Sigma)$, with $\mu(\bar{x}) = \bar{c}$.

(2) ⇒ (1). Consider a specialization $s = (p, V)$ of q. Assume that there exists a Σ-grounding $g_s(\bar{x})$ of s that maps to $\mathsf{chase}_{\downarrow}(D, \Sigma)$ via an injective mapping μ with $\mu(\bar{x}) = \bar{c}$; actually, the fact that μ is injective is not crucial here. By Claim C.4, we get that p maps to

$$\mathsf{chase}(\mathsf{chase}_{\downarrow}(D, \Sigma), \Sigma) \ = \ \mathsf{chase}(D, \Sigma)$$

via a homomorphism ξ with $\xi(\bar{x}) = \bar{c}$. Clearly, by the definition of contractions, there exists a homomorphism h, which is the identity on $\bar{x}$, that maps q to p. Therefore, $\lambda = \xi \circ h$ maps q to $\mathsf{chase}(D, \Sigma)$ with $\lambda(\bar{x}) = \bar{c}$, which implies that $\bar{c} \in q(\mathsf{chase}(D, \Sigma))$. □

C.3 $\mathbb{UCQ}_k$-approximations

We proceed to introduce the notion of $\mathbb{UCQ}_k$-approximation for OMQs from $(\mathbb{G}, \mathbb{UCQ})$, which relies on the notion of grounding of a CQ specialization. Roughly, the $\mathbb{UCQ}_k$-approximation of an OMQ $Q = (\mathbf{S}, \Sigma, q)$, for $k \ge 1$, is the OMQ Q^a_k obtained from Q by replacing each disjunct p of q with the UCQ consisting of all the Σ-groundings of all specializations of p with treewidth at most k.

Definition C.6. Consider an OMQ $Q = (\mathbf{S}, \Sigma, q)$ from $(\mathbb{G}, \mathbb{UCQ})$ over a schema $\mathbf{T}$. The $\mathbb{UCQ}_k$-*approximation* of Q, for $k \ge 1$, is the OMQ $Q^a_k = (\mathbf{S}, \Sigma, q^a_k)$, where q^a_k is obtained from q by replacing each disjunct p of q with the UCQ consisting of all the Σ-groundings over $\mathbf{T}$ of treewidth at most k of all specializations of p. ∎

The main technical lemma concerning $\mathbb{UCQ}_k$-approximations follows. This relies on the notion of k-unraveling of a database D up to a certain tuple $\bar{c}$ over $\mathsf{dom}(D)$, which can be defined in the same way as in [7]. In particular, the *k-unraveling of D up to $\bar{c}$*, for some $k \ge \mathsf{ar}(\mathbf{S}) - 1$, denoted $D^k_{\bar{c}}$, is a possibly infinite $\mathbf{S}$-instance for which the following properties hold:

(1) The subgraph of $G^{D^k_{\bar{c}}}$ induced by the elements of $\mathsf{dom}(D^k_{\bar{c}}) \setminus \bar{c}$ has treewidth k.

(2) $D^k_{\bar{c}} \to D$ via a homomorphism that is the identity on $\bar{c}$.
(3) For every OMQ Q from $(\mathbb{G}, \mathbb{UCQ}_k)$ with data schema $\mathbf{S}$, $\bar{c} \in Q(D)$ implies $\bar{c} \in Q(D^k_{\bar{c}})$.

Let us stress that item (3) relies on the assumption that $k \geq \mathsf{ar}(\mathbf{S}) - 1$ because, otherwise, there can be single atoms in D that cannot be part of any database of treewidth k. In what follows, we say that D has *treewidth k up to $\bar{c}$* if the subgraph of G^D induced by the elements of $\mathsf{dom}(D) \setminus \bar{c}$ has treewidth k.

Lemma C.7. *Let Q be an OMQ from $(\mathbb{G}, \mathbb{UCQ})$ over $\mathbf{T}$ with data schema $\mathbf{S} \subseteq \mathbf{T}$, and Q^a_k its $\mathbb{UCQ}_k$-approximation for $k \geq \mathsf{ar}(\mathbf{T}) - 1$.*

(1) $Q^a_k \subseteq Q$.
(2) For every $\mathbf{S}$-database D and tuple $\bar{c}$ over $\mathsf{dom}(D)$ such that D has treewidth at most k up to $\bar{c}$, $\bar{c} \in Q(D)$ implies $\bar{c} \in Q^a_k(D)$.
(3) For every OMQ $Q' \in (\mathbb{G}, \mathbb{UCQ}_k)$ with data schema $\mathbf{S}$ such that $Q' \subseteq Q$, it holds that $Q' \subseteq Q^a_k$.

Proof. Let $Q = (\mathbf{S}, \Sigma, q(\bar{x}))$, and $Q^a_k = (\mathbf{S}, \Sigma, q^a_k(\bar{x}))$.

Item (1). Consider an $\mathbf{S}$-database D, and a tuple $\bar{c} \in \mathsf{dom}(D)^{|\bar{x}|}$. Assume that $\bar{c} \in Q^a_k(D)$, or, equivalently, there is a disjunct p^a_k of q^a_k that maps to $\mathsf{chase}(D, \Sigma)$ via a homomorphism μ with $\mu(\bar{x}) = \bar{c}$. By definition, p^a_k is a Σ-grounding of a specialization $s = (p_c, V)$ of some disjunct p of q. By Claim C.4, p_c maps to

$$\mathsf{chase}(\mathsf{chase}(D, \Sigma), \Sigma) \;=\; \mathsf{chase}(D, \Sigma)$$

via a homomorphism μ' with $\mu'(\bar{x}) = \bar{c}$. Since, by definition of contractions, there exists a homomorphism h, which is the identity on $\bar{x}$, that maps p to p_c, we get that $\lambda = \mu' \circ h$ maps p to $\mathsf{chase}(D, \Sigma)$ with $\lambda(\bar{x}) = \bar{c}$. Since p is a disjunct of q, we get that $\bar{c} \in Q(D)$.

Item (2). Consider an $\mathbf{S}$-database D, and a tuple $\bar{c} \in \mathsf{dom}(D)^{|\bar{x}|}$ such that D has treewidth at most k up to $\bar{c}$. Assume that $\bar{c} \in Q(D)$. Thus, there exists a disjunct p of q such that $\bar{c} \in p(\mathsf{chase}(D, \Sigma))$. By Lemma C.5, there is a specialization s of p such that a Σ-grounding $g_s(\bar{x})$ of s maps to $\mathsf{chase}_{\downarrow}(D, \Sigma)$ via an injective mapping μ with $\mu(\bar{x}) = \bar{c}$. Since D has treewidth at most k up to $\bar{c}$, also $\mathsf{chase}_{\downarrow}(D, \Sigma)$ has treewdith at most k up to $\bar{c}$. Consequently, g_s has treewidth at most k, and thus is a disjunct of q^a_k. This implies that $\bar{c} \in Q^a_k(D)$.

Item (3). Let $Q' = (\mathbf{S}, \Sigma', q')$. Consider an $\mathbf{S}$-database D and tuple of constants $\bar{c} \in \mathsf{dom}(D)^{|\bar{x}|}$, and assume that $\bar{c} \in Q'(D)$. Since Q' belongs to $(\mathbb{G}, \mathbb{UCQ}_k)$, we get that $\bar{c} \in Q'(D^k_{\bar{c}})$, and thus, $\bar{c} \in Q(D^k_{\bar{c}})$; recall that $D^k_{\bar{c}}$ is the k-unraveling of D up to $\bar{c}$. From item (2), we get that $\bar{c} \in Q^a_k(D^k_{\bar{c}})$, or, equivalently, there is $p \in q^a_k$ that maps to $\mathsf{chase}(D^k_{\bar{c}}, \Sigma)$ via a homomorphism μ with $\mu(\bar{x}) = \bar{c}$. We also know that $D^k_{\bar{c}}$ maps to D via a homomorphism λ with $\lambda(\bar{c}) = \bar{c}$, which allows us to show that $\mathsf{chase}(D^k_{\bar{c}}, \Sigma)$ maps to $\mathsf{chase}(D, \Sigma)$ via an extension λ' of λ. Therefore, $\xi = \lambda' \circ \mu$ maps p to $\mathsf{chase}(D, \Sigma)$ with $\xi(\bar{x}) = \bar{c}$, and thus, $\bar{c} \in Q^a_k(D)$, as needed. □

C.4 Finalizing the Proof of Proposition 5.2

Having Lemma C.7 in place, it is now easy to establish Proposition 5.2. Observe that (3) ⇒ (2) and (2) ⇒ (1) hold trivially. It remains to show that (1) ⇒ (3). By hypothesis, there exists an OMQ Q' from $(\mathbb{G}, \mathbb{UCQ}_k)$ such that $Q \equiv Q'$. By item (3) of Lemma C.7, $Q' \subseteq Q^a_k$, which implies that $Q \subseteq Q^a_k$. By item (1) of Lemma C.7, we also get that $Q^a_k \subseteq Q$, and the claim follows.

C.5 The Case $k < \mathsf{ar}(\mathbf{T}) - 1$

Notice that Theorem 5.1 and Proposition 5.2 only cover the case where $k \geq \mathsf{ar}(\mathbf{T}) - 1$, and also Lemma C.7, which gives the main technical properties of $\mathbb{UCQ}_k$-approximations, requires this condition. We view the case $k < \mathsf{ar}(\mathbf{T}) - 1$ as a somewhat esoteric corner case. In the context of Theorem 5.1, for example, we ask for an equivalent OMQ from $(\mathbb{G}, \mathbb{UCQ}_k)$ when the arity of some relations R is so high that the actual UCQ in such an OMQ cannot contain a fact $R(x_1, \ldots, x_n)$ unless some of the x_i, x_j are identical. We show that this case is also technically very different. In particular, our $\mathbb{UCQ}_k$-approximations do not serve their purpose.

We consider the case where $k = 1$, while $\mathsf{ar}(\mathbf{S}) = 3$ and $\mathsf{ar}(\mathbf{T}) = 6$ with $\mathbf{S}$ being the data schema and $\mathbf{T}$ being the extended schema. Let $\mathbf{S} = \{T_1, T_2\}$ with both T_1 and T_2 ternary. We use the Boolean CQ (for brevity, the existential quantifiers are omitted)

$$q() = R(x_1, x_2) \wedge R(x_1, x_3) \wedge R(x_2, x_4) \wedge R(x_3, x_4) \wedge A_1(x_2) \wedge A_2(x_3)$$

which is a core of treewidth $2 > k$. Our ontology takes the form

$$\Sigma = \{T_1 \to q()\} \cup \Sigma_1 \cup \Sigma_2$$

where Σ_1 makes sure that there is an S-path of length 2^n whenever there is a T_1-atom and Σ_2 make sure that there is an S-path of length $2^n - 1$ whenever there is a T_2-atom, where S is a binary relation symbol. We need auxiliary relation symbols to establish these paths, but we will make sure that they are of arity at least three so that they cannot be 'seen' by UCQ_1-approximations. Note that while the presence of a T_1-atom implies that $q()$ is true, the presence of a T_2-atom does not. In detail, Σ_1 contains the following TGDs:

- $T_1(x_1, x_2, x_3) \to B^0_i(x_1, x_2, x_3)$ for $1 \leq i \leq n$;
- for $1 \leq i \leq n$:

$$B^0_i(x_1, x_2, x_3) \to \exists y_1 \exists y_2 \exists y_3 \, G(x_1, x_2, x_3, y_1, y_2, y_3) \wedge S(x_1, y_1)$$

- for all $i < n$, where $\mathbf{G}$ abbreviates $G(x_1, x_2, x_3, y_1, y_2, y_3)$:

$$\mathbf{G} \wedge \textstyle\bigwedge_{j=0}^{i-1} B^1_j(x_1, x_2, x_3) \wedge B^1_i(x_1, x_2, x_3) \to B^0_j(y_1, y_2, y_3)$$
$$\mathbf{G} \wedge \textstyle\bigwedge_{j=0}^{i-1} B^1_j(x_1, x_2, x_3) \wedge B^0_i(x_1, x_2, x_3) \to B^1_j(y_1, y_2, y_3)$$

- for all i, j with $j < i < n$, using the same abbreviation $\mathbf{G}$:

$$\mathbf{G} \wedge B^0_j(x_1, x_2, x_3) \wedge B^0_i(x_1, x_2, x_3) \to B^0_j(y_1, y_2, y_3)$$
$$\mathbf{G} \wedge B^0_j(x_1, x_2, x_3) \wedge B^1_i(x_1, x_2, x_3) \to B^1_j(y_1, y_2, y_3).$$

We leave the definition of Σ_2 to the reader.

This defines an OMQ $Q = (\mathbf{S}, \Sigma, q)$. It is not hard to see that Q is equivalent to an OMQ Q' from $(\mathbb{G}, \mathbb{UCQ}_1)$, namely to $Q' = (\mathbf{S}, \Sigma, q')$ where $q'()$ is the Boolean CQ that asks for the existence of an S-path of length 2^n. Note that the size of Q' is exponential in the size of Q. This cannot be avoided unless we use an ontology different from Σ, that is, resort to non-uniform UCQ_1-equivalence.

Lemma C.8. *Let $Q'' = (\mathbf{S}, \Sigma, q'') \in (\mathbb{G}, \mathbb{UCQ}_1)$ be equivalent to Q. Then q'' must contain a CQ with at least 2^n atoms.*

Proof. Take the S-database $D_1 = \{T_1(c_1, c_2, c_3)\}$. Then Q evaluates to true on D_1 and thus so does Q''. Consequently, there is a CQ p in q'' and a homomorphism h from p to chase(D_1, Σ). Since p is of treewidth 1, it cannot contain any atoms $P(\bar{x})$ with P of arity at least three and all variables in $\bar{x}$ different, and thus h is a homomorphism from p to chase$(D_1, \Sigma)^-$, which is obtained from chase(D_1, Σ) by removing all facts that use predicates of arity at least three (by choice of D_1 and definition of Σ, none of these facts contains repeated constants) and which takes the form of an S-path of length 2^n. We aim to show that p has at least 2^n atoms. To this end, it suffices to show that for every edge $S(a_1, a_2)$ in the path chase$(D_1, \Sigma) \setminus \{T_1(c_1, c_2, c_3)\}$, there is an atom $S(x_1, x_2) \in p$ such that $h(x_i) = a_i$ for $i \in \{1, 2\}$. Assume that this is not the case. Then cleary we also find a homomorphism from p into an S-path of length $2^n - 1$, thus into chase(D_2, Σ) where $D_2 = \{T_2(c_1, c_2, c_3)\}$. But Q evaluates to false on D_2, which is a contradiction. □

Lemma C.8 allows us to show that the $\mathbb{UCQ}_k$-approximations Q_k^a do not behave as intended when $k < \mathrm{ar}(\mathbf{T}) - 1$. In particular, although the OMQ $Q = (\mathbf{S}, \Sigma, q)$ defined above is $\mathbb{UCQ}_1$-equivalent, we can show that is not equivalent to Q_1^a. By Lemma C.8, we get that for any $Q'' = (\mathbf{S}, \Sigma, q'')$ from $(\mathbb{G}, \mathbb{UCQ}_1)$ that is equivalent to Q, q'' necessarily contains a CQ of exponential size. On the other hand, the $\mathbb{UCQ}_1$-approximation Q_1^a of Q contains only CQs of polynomial size since $\mathrm{ar}(\mathbf{T})$ is a constant. Therefore, Q is not equivalent to Q_1^a.

For $k < \mathrm{ar}(\mathbf{T}) - 1$, we conjecture that every $\mathbb{UCQ}_k$-approximation that is based on the same ontology as the original OMQ Q, and has the fundamental property of being equivalent to Q when the later is uniformly $\mathbb{UCQ}_k$-equivalent, must contain a CQ of double exponential size. This is not true our notion of $\mathbb{UCQ}_k$-approximation in Definition C.6, where each CQ is, in general, of exponential size.

D PROOF OF THEOREM 5.4

D.1 Preliminaries

We start by showing that existential quantifiers can be eliminated:

Theorem D.1. *Consider an OMQ $Q = (\mathbf{S}, \Sigma, q) \in (\mathbb{G}, \mathbb{UCQ})$, where Σ and q are over the schema $\mathbf{T} \supseteq \mathbf{S}$. We can construct an OMQ $Q' = (\mathbf{S}, \Sigma\, q') \in (\mathbb{G} \cap \mathbb{FULL}, \mathbb{UCQ})$, where Σ' and q' are over $\mathbf{T}$, such that, for every $\mathbf{S}$-database D, $Q(D) = Q'(D)$.*

Before giving the proof of the above result, let us recall a key property of linear TGDs, that is, UCQ-rewritability, shown in [15].

Proposition D.2. *Given a set $\Sigma \in \mathbb{L}$ and a UCQ q, both over a schema $\mathbf{T}$, we can construct a UCQ q' over $\mathbf{T}$ such that, for every $\mathbf{T}$-database D, $q(\mathsf{chase}(D, \Sigma)) = q'(D)$.*

We are now ready to give the proof.

Proof. (Theorem D.1) The desired OMQ Q' can be actually extracted from the proof of Lemma A.3. From that proof, we know that we can construct, for some schema $\mathbf{T}'$ disjoint from $\mathbf{T}$,

- a set $\Sigma_1 \in \mathbb{G} \cap \mathbb{FULL}$, over $\mathbf{T}$,
- a set $\Sigma_2 \in \mathbb{G} \cap \mathbb{FULL}$ over $\mathbf{T} \cup \mathbf{T}'$, where the predicates of $\mathbf{T}$ (resp., $\mathbf{T}'$) occur only in the body (resp., head) of a TGD, and
- a set $\Sigma_3 \in \mathbb{L}$ over $\mathbf{T} \cup \mathbf{T}'$,

such that, for every S-database D,

$$Q(D) \;=\; q(\mathsf{chase}(\mathsf{chase}(D, \Sigma_1 \cup \Sigma_2), \Sigma_3)).$$

Notice that we can further assume that $\Sigma_1 \cup \Sigma_2$ consists of TGDs with only one atom in the head; simply split a multi-head full TGD into several single-head TGDs, one for each head atom, that have the same body. Since Σ_3 is set of linear TGDs, by Proposition D.2, we can construct a UCQ $\hat{q}$ over $\mathbf{T} \cup \mathbf{T}'$ such that

$$q(\mathsf{chase}(\mathsf{chase}(D, \Sigma_1 \cup \Sigma_2), \Sigma_3)) \;=\; \hat{q}(\mathsf{chase}(D, \Sigma_1 \cup \Sigma_2)).$$

To obtain the desired set Σ' and UCQ q', it remains to eliminate the predicates of $\mathbf{T}'$. This can be achieved by simply unfolding the atoms in $\hat{q}$ with a predicate from $\mathbf{T}'$ using the single-head TGDs of Σ_2, and get a new UCQ $\tilde{q}$ over $\mathbf{T}$. It is clear that

$$\hat{q}(\mathsf{chase}(D, \Sigma_1 \cup \Sigma_2)) \;=\; \tilde{q}(\mathsf{chase}(D, \Sigma_1)).$$

Therefore, the claim follows with $Q' = (\mathbf{S}, \Sigma_1, \tilde{q})$. □

Let $\mathbf{S}$ be a schema, D an $\mathbf{S}$-database, and $\bar{a} \subseteq \mathsf{dom}(D)$ a guarded set in D. We aim to define a potentially infinite database $D^{\bar{a}}$, the *guarded unraveling of D at $\bar{a}$*. In parallel with $D^{\bar{a}}$, we define a tree decomposition (T, χ), $T = (V, E)$, of the Gaifman graph of $D^{\bar{a}}$ whose width is the maximum arity of relation names in $\mathbf{S}$ minus one.

Let V be the set of sequences $v = \bar{a}_0 \cdots \bar{a}_n$ of guarded sets in D such that $\bar{a}_0 = \bar{a}$ and $\bar{a}_i \cap \bar{a}_{i+1} \neq \emptyset$ for $0 \leq i < n$. Further, let

$$E = \{(v, v') \in V \times V \mid v' = v\bar{b} \text{ for some } \bar{b}\}.$$

We associate with each $v \in V$ a database $D(v)$ and then define $\chi(v)$ to be $\mathsf{dom}(D(v))$. This completes the definition of (T, χ) and allows us to define $D^{\bar{a}}$ as $\bigcup_{v \in V} D(v)$.

Take an infinite supply of *copies* of every $a \in \mathsf{dom}(D)$. Set $b^{\uparrow} = a$ if b is a copy of a, and $a^{\uparrow} = a$. For $v \in V$, define $D(v)$ and its domain $\mathsf{dom}(D(v))$ by induction on the length of the sequence v. For $v = \bar{a}$, $D(v)$ is $D_{|\bar{a}}$, the restriction of D to those atoms with only constants in $\bar{a}$. To define $D(v)$ when $v = \bar{a}_0 \cdots \bar{a}_n$ with $n > 0$, take for any $a \in \bar{a}_n \setminus \bar{a}_{n-1}$ a fresh copy a' of a and define $D(v)$ with domain

$$\begin{aligned} \mathsf{dom}(D(v)) \quad = \quad & \{b \in \mathsf{dom}(D(\bar{a}_0 \cdots \bar{a}_{n-1}) \mid b^{\uparrow} \in \bar{a}_n \cap \bar{a}_{n-1}\} \cup \\ & \{a' \mid a \in \bar{a}_n \setminus \bar{a}_{n-1}\} \end{aligned}$$

such that $b \mapsto b^{\uparrow}$ is an isomorphism from $D(v)$ to $D_{|\bar{a}_n}$.

Injective homomorphisms play a crucial role in our proof of Theorem 5.4. Here, we make some basic observations pertaining to such homomorphisms. For a database D, a CQ $q(\bar{x})$, and a tuple of distinct constants $\bar{a}$, we write $D \models^{io} q(\bar{a})$ if $D \models q(\bar{a})$ and all homomorphisms h from q to D with $h(\bar{x}) = \bar{a}$ are injective. Note that since the constants in $\bar{a}$ are distinct, the condition $h(\bar{x}) = \bar{a}$ does not conflict with injectivity. Here, 'io' stands for 'injectively only'. The following is a simple observation, see [7] for a proof.

Lemma D.3. *If $D \models q(\bar{a})$, for D a potentially infinite database, q a CQ, and $\bar{a}$ a tuple of distinct constants, then $D \models^{io} q_c(\bar{a})$ for some contraction q_c of q.*

The next lemma is an important ingredient to the proof of Theorem 5.4. We write $D, \bar{a} \to E, \bar{b}$ to indicate that there is a homomorphism h from database D to database E with $h(\bar{a}) = \bar{b}$.

Lemma D.4. *Let $Q = (\mathbf{S}, \Sigma, q) \in (\mathbb{G} \cap \mathbb{FULL}, \mathbb{UCQ})$ and $\bar{a}$ a tuple of distinct constants. Then for every $\mathbf{S}$-database D with $D \models Q(\bar{a})$, there is an $\mathbf{S}$-database $\widehat{D}$ such that the following are satisfied:*

(1) $\widehat{D} \models Q(\bar{a})$ and $\widehat{D}, \bar{a} \to D, \bar{a}$;
(2) if $\mathsf{chase}(\widehat{D}, \Sigma) \models^{io} q_c(\bar{a})$, for q_c a contraction of q, then there is no $\mathbf{S}$-database D' and contraction q_c' of q where $D', \bar{a} \to \widehat{D}, \bar{a}$, $\mathsf{chase}(D', \Sigma) \models^{io} q_c'(\bar{a})$, and $q_c \neq q_c'$ is a contraction of q_c'.

Proof. We start with $\widehat{D} = D$. Then, of course, Condition 2 is not guaranteed to be satisfied. We thus iteratively replace $\widehat{D}$ with more suitable databases, as follows. As long as there are an $\mathbf{S}$-database D' and contractions q_c, q_c' of q such that $\mathsf{chase}(\widehat{D}, \Sigma) \models^{io} q_c(\bar{a})$, $D', \bar{a} \to \widehat{D}, \bar{a}$, $\mathsf{chase}(D', \Sigma) \models^{io} q_c'(\bar{a})$, and $q_c \neq q_c'$ is a contraction of q_c', replace $\widehat{D}$ by D'. Informally, this iterative process terminates since we lose at least one contraction q_c of q with $\mathsf{chase}(\widehat{D}, \Sigma) \models^{io} q_c(\bar{a})$ in every step. We refer to [7] for details.

It is clear that the resulting $\widehat{D}$ satisfies Condition (2). Condition (1) is satisfied as well: we have $\mathsf{chase}(\widehat{D}, \Sigma) \models^{io} q_c(\bar{a})$ for some contraction q_c of q, thus $\widehat{D} \models Q(\bar{a})$. □

Lemma D.5. *Given an OMQ $Q = (\mathbf{S}, \Sigma, q)$ from $(\mathbb{G} \cap \mathbb{FULL}, \mathbb{UCQ})$ and an $\mathbf{S}$-database $\widehat{D}$, it is decidable whether Conditions 1 and 2 from Lemma D.4 hold.*

Proof. As Condition 1 is clearly decidable, we concentrate on Condition 2. We start with a central observation. Let q_c' be a contraction of q and let q_c'' be the UCQ that consists of all contractions of q_c' that are distinct from q_c'. Then checking whether $\mathsf{chase}(D', \Sigma) \models^{io} q_c'(\bar{a})$, for some $\mathbf{S}$-database D' with $D', \bar{a} \to \widehat{D}, \bar{a}$, is equivalent to checking that the OMQs $(\mathbf{S}, \Sigma, q_c')$ and $(\mathbf{S}, \Sigma, q_c'')$ are not equivalent on $\mathbf{S}$-databases D' with $D', \bar{a} \to \widehat{D}, \bar{a}$. As a consequence, we can decide Condition 2 by checking that there are no contractions q_c, q_c' of q such that q_c is a contraction of q_c', $q_c \neq q_c'$, $\mathsf{chase}(\widehat{D}, \Sigma) \models q_c(\bar{a})$ and $(\mathbf{S}, \Sigma, q_c'')$ are not equivalent on $\mathbf{S}$-databases D' with $D', \bar{a} \to \widehat{D}, \bar{a}$. Note that in the statement '$\mathsf{chase}(\widehat{D}, \Sigma) \models q_c(\bar{a})$', we have dropped the $\cdot^{io}$ that is present in Condition 2. This is not harmful since if $\mathsf{chase}(\widehat{D}, \Sigma) \models q_c(\bar{a})$, but $\mathsf{chase}(\widehat{D}, \Sigma) \not\models^{io} q_c(\bar{a})$, we can simply replace q_c by a contraction of itself that satisfies this stronger condition.

The *width* of an OMQ $Q = (\mathbf{S}, \Sigma, q)$ from $(\mathbb{G} \cap \mathbb{FULL}, \mathbb{UCQ})$, denoted $\mathsf{w}(Q)$, is the maximum number of variables in the body of a rule in Σ or in a CQ in q.

To establish decidability of Condition 2, it suffices to show that given OMQs Q_1 and Q_2 from $(\mathbb{G} \cap \mathbb{FULL}, \mathbb{UCQ})$ over schema $\mathbf{S}$, an $\mathbf{S}$-database $\widehat{D}$, and a tuple of constants $\bar{a}$, it is decidable whether there exists an $\mathbf{S}$-database D such that $D \models Q_1(\bar{a})$, $D \not\models Q_2(\bar{a})$, and $D, \bar{a} \to \widehat{D}, \bar{a}$. We note that this is the case iff there exists a database with these properties that is of treewidth $\mathsf{w}(Q_1)$.

For an OMQ $Q = (\mathbf{S}, \Sigma, q)$ from $(\mathbb{G} \cap \mathbb{FULL}, \mathbb{UCQ})$ we construct a Guarded Second Order (GSO) [25] formula ϕ_Q such that for every $\mathbf{S}$-database D and tuple of constants $\bar{a}$, $D \models Q(\bar{a})$ iff $D \models \phi_Q(\bar{a})$. Let $J_1, \ldots, J_n$ be the relation symbols in Σ and reserve fresh relation symbols $J_1', \ldots, J_n'$ and $J_1'', \ldots, J_n''$ of the same arity. Further let:

- ϕ_Σ' (resp. ϕ_Σ'') be the conjunction of all TGDs in Σ, viewed as universally quantified FO formulas, with each relation symbol J_i replaced by J_i' (resp. J_i'');
- $\phi_{\mathbf{S}}$ be a formula which says that if J_i is in $\mathbf{S}$, then the extension of J_i is included in that of J_i';
- $\phi_!$ be a formula which says that the extension of each J_i'' is included in the extension of J_i', with at least one inclusion being strict.

Then ϕ_Q is the following formula:[4]

$$\exists J_1', \ldots, J_n' \, (\phi_{\mathbf{S}} \wedge \phi_\Sigma \wedge q \wedge \forall J_1'' \ldots J_n'' \, (\phi_! \to \neg \phi_\Sigma')).$$

Note that the extension of $J_1', \ldots, J_n'$ represents the chase of the input database with Σ.

We also take a formula $\phi_{\widehat{D}}$ that evaluates to true exactly on those $\mathbf{S}$-databases D such that $D, \bar{a} \to \widehat{D}, \bar{a}$. Then, we have to decide whether there exists an $\mathbf{S}$-database D of treewidth at most $\mathsf{w}(Q_1)$ such that $D \models \phi_{Q_1} \wedge \neg \phi_{Q_2} \wedge \phi_{\widehat{D}}$. Note that on sparse structures, GSO has the same expressive power as MSO [13]. As databases of bounded treewidth are sparse [19], it follows that GSO over such databases is decidable. □

For $k, \ell \geq 1$, the *$k \times \ell$-grid* is the (undirected) graph with vertex set $\{(i, j) \mid 1 \leq i \leq k \text{ and } 1 \leq j \leq \ell\}$ and an edge between (i, j) and (i', j') iff $|i - i'| + |j - j'| = 1$. A graph H is a *minor* of a graph G if H is isomorphic to a graph that can be obtained from a subgraph of G by contracting edges. Equivalently, H is a minor of G if there is a *minor map* from $H = (V_H, E_H)$ to $G = (V_E, E_G)$, that is, a mapping $\mu : V_H \to 2^{V_G}$ with the following properties for all $v, w \in V_H$:

- $\mu(v)$ is non-empty and connected in G;
- $\mu(v)$ and $\mu(w)$ are disjoint whenever $v \neq w$;
- if $\{v, w\} \in E_H$, then there are $v' \in \mu(v)$ and $w' \in \mu(w)$ such that $\{v', w'\} \in E_G$.

We say that μ is *onto* if $\bigcup_{v \in V_H} \mu(v) = V_G$. For a database D, we say that $a \in \mathsf{dom}(D)$ is *isolated in D* if there is only a single atom $R(\bar{a}) \in D$ with $a \in \bar{a}$. When k is understood from the context, we use K to denote $\binom{k}{2}$.

Note that Point (3) of the following theorem can be viewed as a form of ontoness requirement for the homomorphism h_0 from Point (1): for the fact $R(a_1, \ldots, a_n) \in D$ mentioned in that condition, it is possible to find a fact $R(c_1, \ldots, c_n) \in D_G$ that maps to it.

Theorem 6.1 (Grohe). *Given an undirected graph G, a $k \geq 1$, a connected $\mathbf{S}$-database D, and a set $A \subseteq \mathsf{dom}(D)$ such that the restriction $G^D_{|_A}$ of the Gaifman graph of D to vertices A contains the $k \times K$-grid as a minor, one can construct in time $f(k) \cdot \mathsf{poly}(||G||, ||D||)$ an $\mathbf{S}$-database D_G with $\mathsf{dom}(D) \setminus A \subseteq \mathsf{dom}(D_G)$ such that:*

(1) there is a surjective homomorphism h_0 from D_G to D that is the identity on $\mathsf{dom}(D) \setminus A$,
(2) G contains a k-clique iff there is a homomorphism h from D to D_G such that h is the identity on $\mathsf{dom}(D) \setminus A$ and $h_0(h(\cdot))$ is the identity, and
(3) if $R(a_1, \ldots, a_n) \in D$, $h_0(b_i) = a_i$ for each $i \in [n]$, and $\{b_i \mid a_i$ non-isolated in $D\}$ is a clique in the Gaifman graph of D_G,

[4] We use the variant of GSO whose the syntax is identical to that of Second Order Logic, but the second-order quantifiers range only over guarded relations, see [25].

then D_G contains an atom $R(c_1, \dots, c_n)$ where $c_i = b_i$ if a_i is non-isolated in D and $h_0(c_i) = a_i$ for $i \in [n]$.

Proof. The construction of D_G is a slight extension of the original construction from [26]. The extension is necessary because we have to treat the constants from $\mathsf{dom}(D) \setminus A$ in a special way whereas in [26] there is no set A or, in other words, $A = \mathsf{dom}(D)$. Let μ be a minor map from the $k \times K$-grid to $G^D_{|A}$. Since D is connected, we can assume w.l.o.g. that μ is onto. To attain Point (3) of Theorem 6.1, we actually have to choose μ a bit more carefully. We first proceed with the construction and argue that Points (1) and (2) are satisfied, and then give details on Point (3). Fix a bijection ρ between $[K]$ and the set of unordered pairs over $[k]$. For $p \in [K]$, let $i \in p$ be shorthand for $i \in \rho(p)$. Let $G^D_{|A} = (V, E)$.

The domain of D_G is

$$(\mathsf{dom}(D) \setminus A) \cup$$
$$\{(v, e, i, p, a) \in V \times E \times [k] \times [K] \times A \mid (v \in e \Leftrightarrow i \in p), a \in \mu(i, p)\}.$$

Before we say what the atoms of D_G are, let us define the homomorphism $h_0 : \mathsf{dom}(D_G) \to \mathsf{dom}(D)$ from Point (1) by taking the identity on $\mathsf{dom}(D) \setminus A$ and the projection to the last component for all other constants from $\mathsf{dom}(D_G)$. We extend h_0 to tuples over $\mathsf{dom}(D_G)$ in the expected way.

We now define D_G to contain every atom $R(\bar{b})$, $\bar{b}$ a tuple over $\mathsf{dom}(D_G)$, such that $R(h_0(\bar{b})) \in D$ and for all constants b, b' in $\bar{a}$ that are of the form $b = (v, e, i, p, a)$ and $b' = (v', e', i', p', a')$,

$$\text{(C1) } i = i' \text{ implies } v = v' \quad \text{and} \quad \text{(C2) } p = p' \text{ implies } e = e'.$$

This finishes the construction of D_G.

Point (1) follows directly from the construction. The proof of Point (2) is a slight variation of the proofs of Lemma 4.3 (for the '$\Rightarrow$' direction) and Lemma 4.4 (for the '$\Leftarrow$' direction) from [26]. In fact, the homomorphism h constructed in the proof of Lemma 4.3 simply needs to be extended to be the identity on $\mathsf{dom}(D) \setminus A$. In Lemma 4.4, the assumption that D is a core has been replaced here with the condition that $h_0(h(\cdots))$ is the identity. This, however, is exactly what is derived from the assumption that D is a core in the original proof. No further changes to the proof are required despite the presence of the elements in $\mathsf{dom}(D) \setminus A$.

For Point (3), assume that $R(a_1, \dots, a_n) \in D$, $h_0(b_j) = a_j$ for $1 \leq j \leq n$, and $S = \{b_j \mid a_j \text{ non-isolated in } D\}$ is a clique in the Gaifman graph of D_G. As D is connected, S cannot be empty. We can assume w.l.o.g that the minor map μ from the $k \times K$-grid onto $G^D_{|A}$ is such that one of the following holds:

(1) there is an a_ℓ that is non-isolated in D such that $a_\ell \in A$ and if i, p are such that $a_\ell \in \mu(i, p)$, then all $a_j \in A$ that are isolated in D are also in $\mu(i, p)$; in this case, b_ℓ must take the form (v, e, i, p, a_ℓ);
(2) there is no a_ℓ that is non-isolated in D such that $a_\ell \in A$ and there are i, p such that all $a_j \in A$ that are isolated in D are in $\mu(i, p)$; in this case, choose v, e arbitrarily such that $v \in e$ iff $i \in p$.

Define $c_1, \dots, c_n$ by setting

$$c_j := \begin{cases} (v, e, i, p, a_j) & \text{if } a_j \in A \text{ is isolated in } D \\ b_j & \text{otherwise.} \end{cases}$$

It can be verified that (v, e, i, p, a_j) is a constant in D_G if $a_j \in A$ is isolated in D, due to our assumptions (1) and (2). Moreover, $h_0(c_i) = h_0(b_i)$ for $1 \leq i \leq n$ and Conditions (C1) and (C2) are satisfied for the fact $R(c_1, \dots, c_n)$, which is thus in D_G. Finally, $h_0(c_i) = a_i$ for $1 \leq i \leq n$, by definition of h_0. □

D.2 The Reduction

Recall that our goal is to establish the following lower bound:

Theorem D.6. *Fix $r \geq 1$. Let $\mathbb{O}$ be a recursively enumerable class of OMQs from $(\mathbb{G}, \mathbb{UCQ})$ over a schema of arity r, and, for each $k \geq 1$, $\mathbb{O} \not\subseteq (\mathbb{G}, \mathbb{UCQ})^{\equiv}_k$. Then,* p-OMQ-Evaluation($\mathbb{O}$) *is* W[1]*-hard.*

The proof is by fpt-reduction from p-Clique, a W[1]-hard problem. Assume that G is an undirected graph and $k \geq 1$ a clique size, given as an input to the reduction. By Robertson and Seymour's Excluded Grid Theorem, there is an ℓ such that every graph of treewidth exceeding ℓ contains the $k \times K$-grid as a minor [34]. We may assume, w.l.o.g., that ℓ exceeds the fixed arity r. By our assumption on $\mathbb{O}$, we find an OMQ $Q = (\mathbf{S}, \Sigma, q)$ from $\mathbb{O}$ that is not in $(\mathbb{G}, \mathbb{UCQ})^{\equiv}_\ell$. Since the choice of Q is independent of G and since it is decidable whether an OMQ from $(\mathbb{G}, \mathbb{UCQ})$ is in $(\mathbb{G}, \mathbb{UCQ})^{\equiv}_\ell$ by Theorem 5.1, we can enumerate the OMQs from $\mathbb{O}$ until we find Q.

By Theorem D.1, we can assume, w.l.o.g., that $Q = (\mathbf{S}, \Sigma, q)$ is from $(\mathbb{G} \cap \mathbb{FULL}, \mathbb{UCQ})$. It might of course be that the OMQ from $(\mathbb{G} \cap \mathbb{FULL}, \mathbb{UCQ})$ is not in $\mathbb{O}$, but we will see that this is not a problem. Let $q = q_1 \vee \cdots \vee q_n$ and let $Q_i = (\mathbf{S}, \Sigma, q_i)$ for $1 \leq i \leq n$. We can assume w.l.o.g. that $Q_i \not\subseteq Q_j$ for all $i \neq j$ because if this is not the case, then we can drop q_j from q in Q and the resulting OMQ is equivalent to Q. Also, we exhaustively replace CQs q_i in Q with $(q_i)^a_\ell$ as defined in the context of $\mathbb{UCQ}_\ell$-approximations (see Definition C.6), whenever the resulting OMQ is equivalent to Q.[5] Since Q is not in $(\mathbb{G}, \mathbb{UCQ})^{\equiv}_\ell$, the final OMQ must contain a q_w that has not been replaced by $(q_w)^a_\ell$ and, in particular, we must have $Q \not\subseteq Q'$ where Q' is obtained from Q by replacing q_w with $(q_w)^a_\ell$. We thus find an $\mathbf{S}$-database D_0 and tuple of constants $\bar{a}_0$ such that $D_0 \models Q_w(\bar{a}_0)$, $D_0 \not\models (Q_w)^a_\ell(\bar{a}_0)$, and $D_0 \not\models Q_i(\bar{a}_0)$ for all $i \neq w$. By duplicating constants, we can achieve that all constants in $\bar{a}_0$ are distinct. We also assume that Condition 2 of Lemma D.4 is satisfied for $Q = Q_w$, $q = q_w$, $\widehat{D} = D_0$, and $\bar{a} = \bar{a}_0$. If it is not, we can apply that lemma and replace D_0 with the resulting $\widehat{D}_0$. By Condition (1), $\widehat{D}_0 \models Q_w(\bar{a}_0)$, $\widehat{D}_0 \not\models (Q_w)^a_\ell(\bar{a}_0)$, and $\widehat{D}_0 \not\models Q_i(\bar{a}_0)$ for all $i \neq w$.

Since the properties of D_0 are independent of G and due to Lemma D.5, we can find D_0 by enumeration. However, D_0 is still not as required and needs to be manipulated further to make it suitable for the reduction. Before we can carry out the actual manipulation, we need some preliminaries.

Recall that for a guarded set $\bar{a} \subseteq \mathsf{dom}(D_0)$, $D_0^{\bar{a}}$ denotes the guarded unraveling of D_0 starting at $\bar{a}$. An *atomic query (AQ)* takes the form $R(\bar{x})$, i.e., it has a single atom and no quantified variables.

Lemma D.7. *$D_0 \models (\mathbf{S}, \Sigma, p)(\bar{a}')$ iff $D_0^{\bar{a}} \models (\mathbf{S}, \Sigma, p)(\bar{a}')$ if p is an AQ and all constants in $\bar{a}'$ are also in $\bar{a}$.*

Of course, $D_0^{\bar{a}}$ can be infinite. By compactness, however, there is a finite $D_{\bar{a}} \subseteq D_0^{\bar{a}}$ such that Lemma D.7 is satisfied for all (finitely

[5] We rely on the key property of $\mathbb{UCQ}_\ell$-approximations given by item (2) of Lemma C.7.

many) AQs p that use only symbols from Σ. For brevity, we say that $D_{\bar{a}}$ *satisfies Lemma D.7 for all relevant AQs*. We can find $D_{\bar{a}}$ by constructing $D_0^{\bar{a}}$ level by level and deciding after each such extension whether we have found the desired database, by checking the condition in Lemma D.7 for all relevant AQs.

A *diversification* of D_0 is a database D obtained from D_0 by replacing every atom $R(\bar{a}) \in D_0$ with a (potentially empty) finite set of atoms $R(\bar{a}'_1), \ldots, R(\bar{a}'_n)$ such that each $\bar{a}'_i$ is obtained from $\bar{a}$ by replacing some constants that do not occur in $\bar{a}_0$ with fresh constants. A constant $a \in \mathsf{dom}(D)$ is *old* if it already occurs in D_0 and *fresh* otherwise. For a fresh constant b, we use $b^\uparrow$ to denote the constant in D_0 that it was introduced for. For old constants a, $a^\uparrow = a$. This extends to tuples of constants from D in a component-wise fashion. Note that all fresh constants in D are isolated in D and that all constants from $\bar{a}_0$ still occur in D as old constants. If D_1 and D_2 are diversifications of D_0, we write $D_1 \preceq D_2$ if there is a homomorphism from D_1 to D_2 that is the identity on all old constants.

Example D.8. Assume that $D = \{R(a,b,c), S(a,b,d)\}$ and $\bar{a}_0 = ()$. Then $D_1 = \{R(a,b',c'), S(a,b,d)\}$ and $D_2 = \{R(a,b,c'), S(a,b,d)\}$ are diversifictions of D and $D_1 \preceq D_2$.

If D is a diversification of D_0, we use D^+ to denote the database obtained from D by adding each database $D_{\bar{a}}$ such that for some atom $R(\bar{a})$ in D, we obtain $D_{\bar{a}}$ from $D_{\bar{a}^\uparrow}$ by renaming the constants in $\bar{a}^\uparrow$ to those in $\bar{a}$. A constant $a \in \mathsf{dom}(D^+)$ is *old* if it is an old constant from $\mathsf{dom}(D)$, and *fresh* otherwise.

For what follows, we choose a $\preceq$-minimal diversification D_1 of D_0 such that $D_1^+ \models Q_w(\bar{a}_0)$.

Example D.9. Assume that q_w is a Boolean CQ that takes the form of an $n \times m$-grid using the binary relations X and Y. Let $\mathbf{S} = \{X', Y'\}$ with both relations ternary and let $\Sigma = \{X'(x,y,z) \to X(x,y),\ Y'(x,y,z) \to Y(x,y)\}$. Further let D_0 be the database

$$\{X'(a_{i,j}, a_{i,j+1}, b) \mid 1 \le i \le m \text{ and } 1 \le j < n\} \cup$$
$$\{Y'(a_{i,j}, a_{i+1,j}, b) \mid 1 \le i < m \text{ and } 1 \le j \le n\}$$

Then the following database is a $\preceq$-minimal diversification of D_0 such that $D_1^+ \models Q_w(\bar{a}_0)$:

$$\{X'(a_{i,j}, a_{i,j+1}, b_{ij}) \mid 1 \le i \le m \text{ and } 1 \le j < n\} \cup$$
$$\{Y'(a_{i,j}, a_{i+1,j}, b'_{ij}) \mid 1 \le i < m \text{ and } 1 \le j \le n\}.$$

We next observe some basic properties of D_1.

Lemma D.10.

(1) $D_1^+ \models Q_w(\bar{a}_0)$ and $D_1^+ \not\models Q_i(\bar{a}_0)$ for all $i \neq w$;
(2) D_1 has treewidth exceeding ℓ up to $\bar{a}_0$;
(3) D_1^+ satisfies Condition 2 of Lemma D.4.

Proof. For Point 1, we have $D_1^+ \models Q_w(\bar{a}_0)$ by choice of D_1^+. By construction of D_1^+, we have $D_1^+, \bar{a}_0 \to D_0, \bar{a}_0$ and thus $D_1^+ \not\models Q_i(\bar{a}_0)$ for all $i \neq w$. We proceed with Point 2, noting that we can argue in the same way to show that $D_1^+ \not\models (Q_w)^a_\ell(\bar{a}_0)$. Thus Point 2 of Lemma C.7 implies that the treewidth of (D_1^+) up to $\bar{a}_0$ exceeds ℓ. Note that the treewidth of $D_1^+ \setminus D_1$ is bounded by the fixed arity r.[6] As we assume $\ell > r$, D_1 has treewidth exceeding ℓ up to $\bar{a}_0$.

Now for Point 3, that is, D_1^+ satisfies Condition 2 of Lemma D.4. Assume to the contrary that there are an $\mathbf{S}$-database D' and contractions q_c, q'_c of q_w such that $\mathsf{chase}(D_1^+, \Sigma) \models^{io} q_c(\bar{a}_0)$, $D', \bar{a}_0 \to D_1^+, \bar{a}_0$, $\mathsf{chase}(D', \Sigma) \models^{io} q'_c(\bar{a}_0)$, and $q_c \neq q'_c$ is a contraction of q'_c. We use a case distinction:

- $\mathsf{chase}(D_0, \Sigma) \models^{io} q_c(\bar{a}_0)$.
  From $D', \bar{a}_0 \to D_1^+, \bar{a}_0$ and $D_1^+, \bar{a}_0 \to D_0, \bar{a}_0$, we obtain $D', \bar{a}_0 \to D_0, \bar{a}_0$. This together with $\mathsf{chase}(D_0, \Sigma) \models^{io} q_c(\bar{a}_0)$ and $\mathsf{chase}(D', \Sigma) \models^{io} q'_c(\bar{a}_0)$ yields a contradiction to D_0 satisfying Condition 2 of Lemma D.4.
- $\mathsf{chase}(D_0, \Sigma) \not\models^{io} q_c(\bar{a}_0)$.
  From $\mathsf{chase}(D_1^+, \Sigma) \models^{io} q_c(\bar{a}_0)$ and $D_1^+, \bar{a}_0 \to D_0, \bar{a}_0$, we obtain $\mathsf{chase}(D_0, \Sigma) \models q_c(\bar{a}_0)$. By Lemma D.3, there is a contraction q'_c of q_c with $\mathsf{chase}(D_0, \Sigma) \models^{io} q'_c(\bar{a}_0)$. But then we must have $q'_c = q_c$ as otherwise we obtain a contradiction to D_0 satisfying Condition 2 of Lemma D.4 (instantiated with $D = \widehat{D} = D_0$). Contradiction.

This completes the proof of Lemma D.10. □

The next lemma states a more intricate property of D_1 that is crucial for the remaining proof.

Lemma D.11. *If h is a homomorphism from $q_w(\bar{x}_0)$ to $\mathsf{chase}(D_1, \Sigma)$ with $h(\mathbf{x}) = \bar{a}_0$, $R(\bar{a}) \in D_1$, and a_1, a_2 are old constants in $\bar{a}$, then there is a path $x_1, \ldots, x_m$ in the Gaifman graph of $D[q_w]$ such that $h(x_1) = a_1$, $h(x_m) = a_2$, and $h(x_2), \ldots, h(x_{m-1})$ are in $\mathsf{dom}(D_{\bar{a}})$.*[7]

Proof. Assume to the contrary that there is a homomorphism h from $q_w(\bar{x}_0)$ to $\mathsf{chase}(D_1^+, \Sigma)$ with $h(\bar{x}_0) = \bar{a}_0$, an $R(\bar{a}) \in D_1$, and old constants a_1, a_2 in $\bar{a}$ such that there is no path in the Gaifman graph $G^{D[q_w]}$ of $D[q_w]$ as required by Lemma D.11. We argue that we can then find an $\mathbf{S}$-database D_2 with $D_2 \prec D_1$ and $D_2^+ \models Q_w(\bar{a}_0)$, in contradiction to the choice of D_1.

Let D_2 be obtained from D_1 by replacing $R(\bar{a})$ with the atoms $R(\bar{a}_1), R(\bar{a}_2)$ where $\bar{a}_i$ is obtained from $\bar{a}$ by replacing a_i with a fresh constant b_i and each fresh constant with a new fresh constant. Clearly, D_2 is a diversification of D_0 and $D_2 \prec D_1$. We convert h into a homomorphism h' from q_w to $\mathsf{chase}(D_2^+, \Sigma)$ such that $h'(\bar{x}) = \bar{a}_0$, which contradicts the choice of D_1.

As a preparation, let us discuss the construction of D_1^+ and D_2^+. During the construction of D_1^+, we have attached a copy $D_{\bar{a}}$ of $D_{\bar{a}^\uparrow}$ at $\bar{a}$. In the construction of D_2^+, we have attached copies $D_{\bar{a}_1}$ and $D_{\bar{a}_1}$ of the same $D_{\bar{a}^\uparrow}$ at $\bar{a}_1$ and $\bar{a}_2$. The constants in $\mathsf{dom}(D_{\bar{a}}) \setminus \mathsf{dom}(D_1)$ are the only constants in D_1^+ that may not be in D_2^+. For $i \in \{1,2\}$, fix an isomorphism ι_i between $D_{\bar{a}}$ and $D_{\bar{a}_i}$. Note that ι_i is the identity on a_{2-i} and in fact on all old constants in $\bar{a}$ except a_i.

Now define $h'(x)$ as follows, for every variable x in q:

(1) if $h(x) \in \mathsf{dom}(D_1^+) \setminus \mathsf{dom}(D_{\bar{a}})$, then $h'(x) = h(x)$
(2) if $h(x) \in \mathsf{dom}(D_{\bar{a}})$:

[6] This is no longer true of Σ is formulated in $\mathbb{FG}$ and this is in fact the reason why our proof does not extended from $\mathbb{G}$ to $\mathbb{FG}$.
[7] A special case is that there is an atom $S(\bar{x})$ in q_w such that a_1 and a_2 are both in $h(\bar{x})$. Also note that it follows that all old constants are in the range of h.

(a) if $G^{D[q_w]}$ contains a path $y_1, \dots, y_p$ such that $h(y_1) = a_i$ for $i \in \{1,2\}$, $y_p = x$, and $h(x_2), \dots, h(x_{p-1})$ are in $\mathsf{dom}(D_{\bar{a}})$, then $h'(x) = \iota_{2-i}(h(x))$;
(b) if $h(x)$ is still undefined, then $h'(x) = \iota_1(h(x))$.

Note that h' is defined properly: due to our assumption that there is no path as in Lemma D.11, the precondition of 2b cannot be satisfied both for $i = 1$ and for $i = 2$. Also note that if $S(\bar{x}) \in q_w$ with $h(\bar{x}) \subseteq \mathsf{adom}(D_{\bar{a}})$, then $h'(x) = \iota_i(h(x))$, for some $i \in \{1,2\}$.

It remains to show that h' is a homomorphism from q_w to $\mathsf{chase}(D_2^+, \Sigma)$. Let us first observe three helpful claims. Recall that $\cdot^\uparrow$ maps each constant in a diversification of D_0, such as D_1 and D_2, to the original constant that it was introduced for. The first claim is a consequence of Lemma D.7 and the fact that since D_1 and D_2 are diversifications of D_0, $\cdot^\uparrow$ defines a homomorphism from D_j to D_0 each $j \in \{1,2\}$ that can be extended to a homomorphism from D_j^+ to D_0 and further from $\mathsf{chase}(D_j^+, \Sigma)$ to $\mathsf{chase}(D_0, \Sigma)$.

Claim 1. Let $\bar{b}$ be a guarded set in D_j, $j \in \{1,2\}$. Then $R(\bar{b}) \in \mathsf{chase}(D_j^+, \Sigma)$ iff $R(\bar{b}^\uparrow) \in \mathsf{chase}(D_0^+, \Sigma)$ for all facts $R(\bar{b})$.

Here is the second claim.

Claim 2. Let $\bar{b}$ be a guarded set in $D_{\bar{a}}$. Then $R(\bar{b}) \in \mathsf{chase}(D_1^+, \Sigma)$ implies $R(\iota_i(\bar{b})) \in \mathsf{chase}(D_2^+, \Sigma)$ for all facts $R(\bar{b})$ and $i \in \{1,2\}$.

Assume to the contrary that $R(\iota_i(\bar{b})) \notin \mathsf{chase}(D_2^+, \Sigma)$. Set

$$D \quad = \quad \mathsf{chase}(D_1^+, \Sigma)|_{\mathsf{dom}(D_1^+) \setminus \mathsf{dom}(D_{\bar{a}})} \cup \{R(\iota_i^-(\bar{c})) \mid R(\bar{c}) \in \mathsf{chase}(D_2^+, \Sigma)|_{\mathsf{dom}(D_{\bar{a}_i})}\}.$$

Using the fact that Σ is guarded, it can be shown that D is a model of Σ and of D_1^+. Moreover, $R(\bar{b}) \notin D$ and, consequently, $R(\bar{b}) \notin \mathsf{chase}(D_1^+, \Sigma)$, which finishes the proof of the claim.

The proof of the third claim is identical to that of Claim 2.

Claim 3. Let $\bar{c}$ be a guarded set in some tree $D_{\bar{b}}$ in D_1^+, $\bar{b} \neq \bar{a}$. Then $R(\bar{c}) \in \mathsf{chase}(D_1^+, \Sigma)$ implies $R(\bar{c}) \in \mathsf{chase}(D_2^+, \Sigma)$ for all atoms $R(\bar{c})$.

Now for the proof that h' is a homomorphism from q_w to $\mathsf{chase}(D_2^+, \Sigma)$. Let $S(\bar{x}) \in q_w$.

First assume that $h(\bar{x})$ contains only constants from $\mathsf{dom}(D_1)$. It follows from (two applications of) Claim 1 that $S(h(\bar{x})) \in \mathsf{chase}(D_1^+, \Sigma)$ implies $S(h'(\bar{x})) \in \mathsf{chase}(D_2^+, \Sigma)$.

Next assume that $h(\bar{x})$ contains a constant from a tree $\mathsf{dom}(D_{\bar{c}})$, $\bar{c} \neq \bar{a}$, that is not in $\mathsf{dom}(D_1)$. By construction of D_1^+ and by definition of the chase, this implies that $h(\bar{x}) \subseteq \mathsf{adom}(D_{\bar{c}})$. We can thus apply Claim 3 to infer that $S(h(\bar{x})) \in \mathsf{chase}(D_1^+, \Sigma)$ implies $S(h'(\bar{x})) \in \mathsf{chase}(D_2^+, \Sigma)$.

Finally assume that $h(\bar{x})$ contains a constant from $\mathsf{dom}(D_{\bar{a}}) \setminus \mathsf{dom}(D_1)$. Then $h(\bar{x}) \subseteq \mathsf{adom}(D_{\bar{a}})$. Thus $h'(\bar{x}) = \iota_i(h(\bar{x}))$ for some $i \in \{1,2\}$. We can thus apply Claim 2 to infer that $S(h(\bar{x})) \in \mathsf{chase}(D_1^+, \Sigma)$ implies $S(h'(\bar{x})) \in \mathsf{chase}(D_2^+, \Sigma)$. □

Clearly, neither D_1 nor D_1^+ are guaranteed to be connected. In particular, D_0 might be disconnected and even if it is connected the diversification might replace all constants in an atom with fresh constants and in this way make D_1 disconnected. By Point 2 of Lemma D.10, however, the restriction of G^{D_1} to $\mathsf{dom}(D_1) \setminus \bar{a}_0$ has a maximal connected component that is of treewidth exceeding ℓ. Let A be the set of constants in that component. Moreover, let D be the maximal connected component of D_1 that contains all constants from A (it might additionally contain constants from $\bar{a}_0$) and let D_{dis}^+ denote the part of D_1^+ that is disconnected from $D \subseteq D_1^+$. By choice of D, A, and ℓ, the restriction of G^D to A contains the $k \times K$-grid as a minor. We can thus apply Theorem 6.1 to G, k, and D, and A, obtaining an S-database D_G and a homomorphism h_0 from D_G to D such that Points 1 to 3 of that theorem are satisfied. Let

(1) D_G^+ be obtained by starting with D_G and then disjointly adding, for each guarded set $\bar{a}$ in D_G such that the restriction of h_0 to $\bar{a}$ is injective, a copy of the database $D_{h_0(\bar{a})}$ that was attached to $R(h_0(\bar{a})) \in D_1$ during the construction of D_1^+, identifying the root of this copy with $\bar{a}$;[8]
(2) D_G^* be obtained by disjointly adding D_{dis}^+.

The fpt-reduction of p-Clique then consists in computing Q and D_G^* from G and $k \geq 1$. We mean here the original Q that is guaranteed to be in $\mathbb{O}$ and where Σ is from $\mathbb{G}$ but not necessarily from $\mathbb{G} \cap \mathbb{FULL}$.

D.3 Correctness of the Reduction

We show in the subsequent lemma that $D_G^* \models Q(\bar{a}_0)$ if and only if G has a k-clique. Clearly, we can work with the version of Q here in which Σ is from $\mathbb{G} \cap \mathbb{FULL}$ since it is equivalent to the original Q.

Lemma D.12. *G has a k-clique iff $D_G^* \models Q(\bar{a}_0)$.*

Proof. The 'only if' direction is easy. If G has a k-clique, then by Point 2 of Theorem 6.1 and since A contains none of the constants from $\bar{a}_0$, there is a homomorphism h from D to D_G such that $h(\bar{a}_0) = \bar{a}_0$ and $h_0(h(\cdot))$ is the identity. It is straightforward to extend h to a homomorphism from D_1^+ to $D_G^* = D_G^+ \uplus D_{\mathsf{dis}}^+$; in particular, if $R(\bar{a})$ is an atom in D, then the restriction of h_0 to $\bar{a}$ must be injective since $h_0(h(\cdot))$ is the identity and thus a copy of the subtree that was attached to $h(\bar{a})$ in the construction of D_1^+ from D_1 was attached to $h(\bar{a})$ in the construction of D_G^+ from D_G. Consequently, $D_1^+ \models Q_w(\bar{a}_0)$ implies $D_G^* \models Q_w(\bar{a}_0)$.

For the 'if' direction, assume that $D_G^* \models Q(\bar{a}_0)$. The surjective homomorphism h_0 from D_G to D given by Theorem 6.1 can be extended to a homomorphism from $D_G^* = D_G^+ \uplus D_{\mathsf{dis}}^+$ to D_1^+ and from $\mathsf{chase}(D_G^*, \Sigma)$ to $\mathsf{chase}(D_1^+, \Sigma)$.[9] For brevity, we denote this extension also with h_0. For later use, we observe that

(P1) $b \in \mathsf{dom}(D_G)$ iff $h_0(b) \in \mathsf{dom}(D)$;
(P2) b_1, b_2 are in the same tree of D_G^+ iff $h_0(b_1), h_0(b_2)$ are in the same tree of D_1^+.

So h_0 witnesses $D_G^*, \bar{a}_0 \to D_1^+, \bar{a}_0$. Thus $D_1^+ \not\models Q_i(\bar{a}_0)$ for all $i \neq w$ implies $D_G^* \not\models Q_i(\bar{a}_0)$ for all $i \neq w$. It follows that $D_G^* \models Q_w(\bar{a}_0)$ and thus $\mathsf{chase}(D_G^*, \Sigma) \models q_w(\bar{a}_0)$.

By Lemma D.3, we find a contraction q_c of q_w such that $\mathsf{chase}(D_G^*, \Sigma) \models^{io} q_c(\bar{a}_0)$. As witnessed by h_0, $\mathsf{chase}(D_G^*, \Sigma), \bar{a}_0 \to \mathsf{chase}(D_1^+, \Sigma), \bar{a}_0$. Consequently $\mathsf{chase}(D_1^+, \Sigma) \models q_c(\bar{a}_0)$. Thus we find a contraction q_c' of q_c such that $\mathsf{chase}(D_1^+, \Sigma) \models^{io} q_c'(\bar{a}_0)$. We must have $q_c = q_c'$ since D_1^+ satisfies Condition 2 of Lemma D.4, via Lemma D.10.

[8] The injectivity requirement implies that we can add these copies as they are, without duplicating any constants in them.
[9] The extension needs not be surjective.

Let h be a homomorphism from q_c to $\mathsf{chase}(D_G^*, \Sigma)$. The composition $g = h_0(h(\cdot))$ is a homomorphism from q_c to $\mathsf{chase}(D_1^+, \Sigma)$. Since $\mathsf{chase}(D_1^+, \Sigma) \models^{io} q_c$, this homomorphism must be injective. Consequently, also h_0 is injective.

Since h_0 is injective, we can construct a function h_0^- as follows:

(1) start with the restriction of h_0 to the range of h;
(2) take the inverse;
(3) restrict to old constants in D.

It follows from Lemma D.11 that the range of h_0 includes all old constants in D, and thus so does the domain of h_0^-. In particular, it thus also contains all constants in $\bar{a}_0$. Trivially, $h_0(h_0^-(\cdot))$ is the identity and h_0^- is the identity on all constants from $\mathsf{dom}(D) \setminus A \subseteq \bar{a}_0$ (since h_0 is the identity on such constants).

We are going to show that a certain extension of h_0^- is a homomorphism from D to D_G that is the identity on $\mathsf{dom}(D) \setminus A$, and thus Point 2 of Theorem 6.1 yields that G contains a k-clique.

Let $R(\bar{a}) \in D$. By definition of g and since the range of g contains all old constants in D, we find for each old $a \in \bar{a}$ a variable $x_a \in \bar{x}$ and a $b_a \in \mathsf{dom}(D_G^*)$ such that $h(x_a) = b_a$ and $h_0(b_a) = a$. Let Γ be the set of all b_a. By Lemma D.11, for any two old constants a_1, a_2 in $\bar{a}$, there is a path $x_1, \ldots, x_n$ in the Gaifman graph of $D[q_w]$ such that $g(x_1) = a_1$, $g(x_m) = a_2$, and $g(x_2), \ldots, g(x_{m-1})$ are in $\mathsf{dom}(D_{\bar{a}})$. By Properties P1 and P2, it follows that for any two b_1, b_2 in Γ, there is a path $x_1, \ldots, x_n$ in the Gaifman graph of $D[q_w]$ such that $h(x_1) = b_1$, $h(x_m) = b_2$, and $h(x_2), \ldots, h(x_{m-1})$ are in the same tree of D_G^+. The fact that h_0 maps all elements of Γ to constants in D and Property P1 further imply $b_1, b_2 \in \mathsf{dom}(D_G)$. By construction of D_G^+, there must thus be a fact in D_G that contains both b_1 and b_2 (the one for which the tree was added that $h(x_2), \ldots, h(x_{m-1})$ are in). It follows that Γ forms a clique in the Gaifman graph of D_G. Since (the non-extended) h_0 is surjective, for each fresh $a \in \bar{a}$ we find some $b_a \in \mathsf{dom}(D_G)$ such that $h_0(b_a) = a$. We can now apply Point 3 of Theorem 6.1 to $R(\bar{a})$ and the b_a to obtain an extension of h_0^- to the fresh constants in $\bar{a}$ such that $R(h_0^-(\bar{a})) \in D_G$. Note that the set $\{b_a \mid a \text{ non-isolated in } D\}$ can only contain old constants since all fresh constants are isolated in D; it is thus a subset of Γ and forms a clique in the Gaifman graph of D_G. The extension is such that if a is fresh and covered by the extension, then $h_0^-(a) = c$ for some c with $h_0(c) = a$. Thus, $h_0(h_0^-(\cdot))$ is still the identity and h_0^- is still the identity on $\mathsf{dom}(D) \setminus A$. Also note that this extension can be done independently for all $R(\bar{a}) \in D$ because all fresh constants in D are isolated in D.

Finally, Point 2 of Theorem 6.1 yields that G has a k-clique. □

E PROOF OF PROPOSITION 5.5

Recall that, given two CQSs $S = (\Sigma, q)$ and $S' = (\Sigma, q')$ over $\mathbf{S}$, we write $S \subseteq S'$ if $q(D) \subseteq q'(D)$ for every $\mathbf{S}$-database D that satisfies Σ. We also write $S \equiv S'$ if $S \subseteq S'$ and $S' \subseteq S$. Analogously, we define the notions that consider unrestricted (not necessarily finite) instances. We write $S \subseteq^{\mathsf{unr}} S'$ if $q(I) \subseteq q'(I)$ for every (possibly infinite) $\mathbf{S}$-instance I that satisfies Σ, and $S \equiv^{\mathsf{unr}} S'$ if $S \subseteq^{\mathsf{unr}} S'$ and $S' \subseteq^{\mathsf{unr}} S$. The next simple result states that for guarded TGDs the notions $\equiv$ and $\equiv^{\mathsf{unr}}$ coincide, which, unsurprisingly, relies on the fact that guarded TGDs are finitely controllable.

Lemma E.1. *Consider two CQSs $S = (\Sigma, q(\bar{x}))$ and $S' = (\Sigma, q'(\bar{x}'))$ from $(\mathbb{G}, \mathbb{UCQ})$. Then, $S \equiv S'$ iff $S \equiv^{\mathsf{unr}} S'$.*

Proof. The ($\Leftarrow$) direction holds trivially. For the ($\Rightarrow$) direction, assume that $S \not\equiv^{\mathsf{unr}} S'$. There are two cases: (i) $S \not\subseteq^{\mathsf{unr}} S'$, or (ii) $S' \not\subseteq^{\mathsf{unr}} S$. We consider only the first case; the second case is shown analogously. Since $S \not\subseteq^{\mathsf{unr}} S'$, we get that $\bar{x} \notin q'(\mathsf{chase}(q, \Sigma))$. With Q' being the OMQ $\mathsf{omq}(S')$, we get that $\bar{x} \notin Q'(D[q])$. Since, by Lemma 6.6 and Theorem 6.7, guarded TGDs are finitely controllable, we conclude that there exists a finite model M of $D[q]$ and Σ such that $\bar{x} \notin q'(M)$. However, it holds trivially that $\bar{x} \in q(M)$, and thus, $S \not\subseteq S'$, which in turn implies that $S \not\equiv S'$, as needed. □

Having the above lemma in place, we can now give the proof of Proposition 5.5. Assume that $S = (\Sigma, q)$ is over a schema $\mathbf{S}$. Since, by Proposition 5.2, an OMQ from $(\mathbb{G}, \mathbb{UCQ})$ is $\mathbb{UCQ}_k$-equivalent iff it is uniformly $\mathbb{UCQ}_k$-equivalent, it suffices to show that, for each $k \geq 1$, the following are equivalent:

(1) There is $q' \in \mathbb{UCQ}_k$ over $\mathbf{S}$ such that $S \equiv (\Sigma, q')$.
(2) There is $q' \in \mathbb{UCQ}_k$ over $\mathbf{S}$ such that $\mathsf{omq}(S) \equiv (\mathbf{S}, \Sigma, q')$.

(1) $\Rightarrow$ (2). By hypothesis, there exists $q' \in \mathbb{UCQ}_k$ over $\mathbf{S}$ such that $S \equiv (\Sigma, q')$. It is easy to show that $\mathsf{omq}(S) \equiv (\mathbf{S}, \Sigma, q')$; for brevity, let $Q = \mathsf{omq}(S)$ and $Q' = (\mathbf{S}, \Sigma, q')$. Consider an arbitrary $\mathbf{S}$-database D. Since, by construction, $\mathsf{chase}(D, \Sigma)$ is a model of Σ, and, by Lemma E.1, $S \equiv^{\mathsf{unr}} (\Sigma, q')$, we get that $q(\mathsf{chase}(D, \Sigma)) = q'(\mathsf{chase}(D, \Sigma'))$. Therefore, $Q(D) = Q'(D)$, as needed.

(2) $\Rightarrow$ (1). By hypothesis, there exists $q' \in \mathbb{UCQ}_k$ over $\mathbf{S}$ such that $\mathsf{omq}(S) \equiv (\mathbf{S}, \Sigma, q')$; again, let $Q = \mathsf{omq}(S)$ and $Q' = (\mathbf{S}, \Sigma, q')$. Consider an arbitrary $\mathbf{S}$-database D such that D is a model of Σ. Observe that $q(D) = Q(D)$ and $q'(D) = Q'(D)$. Since $Q \equiv Q'$, we get that $q(D) = q'(D)$, and the claim follows.

F PROOF OF PROPOSITION 5.8

Consider an OMQ $Q = (\mathbf{S}, \Sigma, q(\bar{x}))$ from $\mathsf{omq}(\mathbb{O})$, an $\mathbf{S}$-database D, and a tuple $\bar{c} \in \mathsf{dom}(D)^{|\bar{x}|}$. We are going to construct an $\mathbf{S}$-database D^* such that $D^* \models \Sigma$, and $\bar{c} \in Q(D)$ iff $\bar{c} \in q(D^*)$, or, equivalently (due to Proposition 2.2), $\bar{c} \in \mathsf{chase}(D, \Sigma)$ iff $\bar{c} \in q(D^*)$.

F.1 Construction of the Database D^*

Note that the definition of D^* has been already given in the main body of the paper (Section 6.2). However, we repeat it here for the sake of readability. We first define D^+ as the database

$$D \cup \{R(\bar{a}) \in \mathsf{chase}(D, \Sigma) \mid \bar{a} \subseteq \mathsf{dom}(D)\}.$$

Let A be the family of all maximal tuples $\bar{a}$ over $\mathsf{dom}(D)$ that are guarded in D^+, i.e., there is an atom $R(\bar{b}) \in D^+$ such that $\bar{a} \subseteq \bar{b}$. Fix an arbitrary tuple $\bar{a} \in A$. Since, by Theorem 6.7, the class $\mathbb{G}$ is strongly finitely controllable, and also finite witnesses are realizable, we can compute an instance $M(D^+_{|\bar{a}}, \Sigma, n) \in \mathsf{fmods}(D^+_{|\bar{a}}, \Sigma)$, where n is the number of variables in q, such that for each CQ q' of arity $|\bar{a}|$ with at most n variables, it holds that

$$(*)\ \ \bar{a} \in q'(M(D^+_{|\bar{a}}, \Sigma, n)) \implies \bar{a} \in q'(\mathsf{chase}(D^+_{|\bar{a}}, \Sigma)).$$

W.l.o.g., we assume that $\mathsf{dom}(M(D^+_{|\bar{a}}, \Sigma, n)) \cap \mathsf{dom}(M(D^+_{|\bar{b}}, \Sigma, n)) \subseteq \mathsf{dom}(D)$, for every two distinct tuples $\bar{a}, \bar{b} \in A$. The database D^* is

$$D^+ \cup \bigcup_{\bar{a} \in A} M(D^+_{|\bar{a}}, \Sigma, n).$$

F.2 Correctness of the Reduction

The next lemma, which is actually Lemma 6.8 in the main body of the paper, shows that D^* is the desired database.

Lemma F.1. *It holds that:*

(1) $D^* \models \Sigma$.
(2) $\bar{c} \in \mathsf{chase}(D, \Sigma)$ *iff* $\bar{c} \in q(D^*)$.
(3) There exists a computable function $f : \mathbb{N} \to \mathbb{N}$ *such that* D^* *can be constructed in time* $||D||^{O(1)} \cdot f(||Q||)$.

Proof. Item (1). Consider a TGD $\sigma \in \Sigma$ of the form $\phi(\bar{x}, \bar{y}) \rightarrow \exists \bar{z}\, \psi(\bar{x}, \bar{z})$, with $\mathsf{guard}(\sigma) = R(\bar{u})$. Let $q_\phi = \exists \bar{y}\, \phi(\bar{x}, \bar{y})$ and $q_\psi = \exists \bar{z}\, \psi(\bar{x}, \bar{z})$. Fix an arbitrary tuple $\bar{c} \in \mathsf{dom}(D^*)^{|\bar{x}|}$, and assume that $\bar{c} \in q_\phi(D^*)$. We need to show that $\bar{c} \in q_\psi(D^*)$. Since $\bar{c} \in q_\phi(D^*)$, we get that $\phi(\bar{x}, \bar{y}) \rightarrow D^*$ via a homomorphism h such that $h(\bar{x}) = \bar{c}$. Clearly, $R(h(\bar{u})) \in M(D_{|\bar{a}}, \Sigma, n)$ for some tuple $\bar{a} \in A$ that contains all the constants $h(\bar{u}) \cap \mathsf{dom}(D)$. It is then easy to see that, due to guardedness, $h(\phi(\bar{x}, \bar{y})) \subseteq M(D_{|\bar{a}}, \Sigma, n)$. Since $M(D_{|\bar{a}}, \Sigma, n)$ is model of Σ, we get that $\bar{c} \in q_\psi(M(D_{|\bar{a}}, \Sigma, n))$. Therefore, due to the monotonicity of CQs, $\bar{c} \in q_\psi(D^*)$, as needed.

Item (2). Let us first concentrate on the ($\Rightarrow$) direction. Since $D^* \models \Sigma$, we get that $\mathsf{chase}(D, \Sigma) \rightarrow D^*$ via a homomorphism h that is the identity on $\mathsf{dom}(D)$; the latter holds due to Proposition 2.2. Moreover, by hypothesis, $q \rightarrow \mathsf{chase}(D, \Sigma)$ via a homomorphism μ that maps the answer variables $\bar{x}$ of q to $\bar{c}$. Therefore, $\mu \circ h$ maps q to D^* and $\bar{x}$ to $\bar{c}$, which in turn implies that $\bar{c} \in q(D^*)$. We now show the ($\Leftarrow$) direction. By hypothesis, $q \rightarrow D^*$ via a homomorphism h such that $h(\bar{x}) = \bar{c}$. For each tuple $\bar{a} \in A$, let $q_{\bar{a}}(\bar{x}_{\bar{a}})$ be the maximal subquery of q such that $h(q_{\bar{a}}) \subseteq M(D_{|\bar{a}}, \Sigma, n)$ and $h_{\bar{a}}(\bar{x}_{\bar{a}}) = \bar{a}$, i.e., $\bar{x}_{\bar{a}}$ are the variables of $q_{\bar{a}}$ that are mapped to constants of $\mathsf{dom}(D)$. By ($*$), we get that $q_{\bar{a}} \rightarrow \mathsf{chase}(D_{|\bar{a}}, \Sigma)$ via a homomorphism $h_{\bar{a}}$ that maps $\bar{x}_{\bar{a}}$ to $\bar{a}$. It should be clear that

$$\mu = \bigcup_{\bar{a} \in A} h_{\bar{a}}$$

is well-defined mapping that maps q to $\mathsf{chase}(D, \Sigma)$ with $\mu(\bar{x}) = \bar{c}$. This implies that $\bar{c} \in q(\mathsf{chase}(D, \Sigma))$, and the claim follows.

Item (3). This is shown by exploiting the fact that the database D^+ can be constructed in time $||D||^{O(1)} \cdot g(||Q||)$ for some computable function $g : \mathbb{N} \to \mathbb{N}$, and the fact that the cardinality of A, as well as the size of $D^+_{|\bar{a}}$ for some $\bar{a} \in A$, do not depend on D. For showing that D^+ can be constructed in the claimed time, we exploit Lemma A.4, and the fact that we can construct a set $\xi(\Sigma) \in \mathbb{G} \cap \mathbb{FULL}$ such that $D^+ = \mathsf{chase}(D, \xi(\Sigma))$; the latter is inherited from [24]. □

G PROOF OF PROPOSITION 5.11

Let $S = (\Sigma, q(\bar{x}))$ be a CQS from $(\mathbb{FG}_m, \mathbb{UCQ})$ over a schema $\mathbf{S}$ of arity r. Via a proof similar to that of Proposition 5.5, we get that:

- S is uniformly $\mathbb{UCQ}_k$-equivalent iff $\mathsf{omq}(S)$ is.
- $S \equiv S^a_k$ iff $\mathsf{omq}(S) \equiv \mathsf{omq}(S^a_k)$.

Let $Q_S = \mathsf{omq}(S)$ and $Q^k_S = \mathsf{omq}(S^a_k)$. Thus, for showing Proposition 5.11, it suffices to show that the following are equivalent:

(1) Q_S is uniformly $\mathbb{UCQ}_k$-equivalent.
(2) $Q_S \equiv Q^k_S$.

The proof of the next lemma is along the lines of the proof of Lemma C.7. The only difference is that we rely on the following fact: given a database D of treewidth at most k up to $\bar{c}$, $\mathsf{chase}(D, \Sigma)$ has treewdith at most k up to $\bar{c}$. The latter heavily exploits the bound m on the number of head atoms in the TGDs of Σ.

Lemma G.1. *For every OMQ* $Q' = (\mathbf{S}, \Sigma, q')$ *from* $(\mathbb{FG}_m, \mathbb{UCQ})$ *such that* $Q' \subseteq Q_S$, *it holds that* $Q' \subseteq Q^k_S$.

Having Lemma G.1 in place, it is now easy to establish the above equivalence, i.e., Q_S is uniformly $\mathbb{UCQ}_k$-equivalent iff $Q_S \equiv Q^k_S$. Observe that the ($\Leftarrow$) direction holds trivially. It remains to show the ($\Rightarrow$) direction. By hypothesis, there exists an OMQ $Q' = (\mathbf{S}, \Sigma, q')$ from $(\mathbb{FG}_m, \mathbb{UCQ}_k)$ such that $Q_S \equiv Q'$. By Lemma G.1, $Q' \subseteq Q^k_S$, which in turn implies that $Q_S \subseteq Q^k_S$. The fact that $Q^k_S \subseteq Q_S$ holds by construction, and the claim follows.

H PROOF OF THEOREM 5.13

We provide a complete proof of Theorem 5.13 here. Our proof does not use Grohe's database as defined in [26], but a variant of it which we define below. We start by introducing several important notions.

Let $G = (V, E)$ and $H = (V', E')$ be simple graphs. A *minor map* from H to G is a mapping $\mu : V' \rightarrow 2^V$ such that:

(1) For every $v' \in V'$, $\mu(v')$ is nonempty and connected in G.
(2) For every $v', u' \in V'$, with $v' \neq u'$, $\mu(v')$, $\mu(u')$ are disjoint. The sets of the form $\mu(v')$, for $v' \in V'$, are pairwise disjoint.
(3) For every edge $\{v', u'\} \in E'$, there are nodes $v \in \mu(v')$ and $u \in \mu(u')$ such that $\{v, u\} \in E$.

Such a minor map is said to be *onto* if, in addition, $\bigcup_{v' \in V'} \mu(v') = V$.

Let G and H be simple graphs. It is easy to see that H is a minor of G iff there is a minor map μ from H to G. If, in addition, G is connected, then there exists an onto minor map μ from H to G.

H.1 A Variation of Grohe's Database

Let $G = (V, E)$ be an undirected graph, $k \geq 1$ be an integer, and $K = \binom{k}{2}$. Suppose that D and D' are $\mathbf{S}$-databases with $D \subseteq D'$ such that there exists a set $A \subseteq \mathsf{dom}(D)$ for which there is a minor map μ from the $(k \times K)$-grid onto $G^D_{|A}$. We fix a bijection χ that assigns to each 2-element subset of $[k]$ a value in $[K]$. We define a database $D^* := D^*(G, D, D', A, \mu)$ as follows.

The domain. Every element of D^* is either an element in $\mathsf{dom}(D') \setminus A$ or a tuple of the form (v, e, i, p, z) where $v \in V$, $e \in E$, $i \in [k]$, p is a two-element subset of $[k]$, and $z \in A$. We note here that $\mathsf{dom}(D^*)$ does not necessarily contain all elements of the previous form since some of them might not appear in any of its atoms.

The facts. A *labelled clique* in G is any partial mapping η from $[k]$ to V such that, for every different i, j in the domain of η, we have that $\eta(i)$ and $\eta(j)$ are adjacent in G. We say that an element $z \in \mathsf{dom}(D')$ is *covered* by a labelled clique η if $z \notin A$, or there exist i, j, ℓ in domain of η such that $z \in \mu(i, \chi(\{j, \ell\}))$. Then, for every fact $R(\bar{z}) \in D'$ and every labelled clique η

covering every element mentioned in $\bar{z}$, it is the case that D^* contains the fact $R(\bar{z}_\eta)$ where $\bar{z}_\eta$ is obtained by replacing, in $\bar{z}$, every element $z \in A$ with

$$(\eta(i), \{\eta(j), \eta(\ell)\}, i, \{j, \ell\}, z),$$

where $\mu(i, \chi\{j, \ell\}) = z$.

Additionally, we define the projection

$$h_0 \,:\, \mathsf{dom}(D^*) \to \mathsf{dom}(D')$$

such that

$$\begin{cases} h_0(v, e, i, p, z) = z, & \text{if } z \in A, \text{ and} \\ h_0(y) = y, & \text{otherwise.} \end{cases}$$

Although we avoid to introduce Grohe's original database [26], it is useful to compare it with ours. In particular, consider the case when $D = D'$ and $A = \mathsf{dom}(D)$, and thus we have a database $D^* = D(G, D, D, \mathsf{dom}(D), \mu)$. Grohe's database $D_{\text{Grohe}} = D_{\text{Grohe}}(G, D, \mu)$ then satisfies the following:

- $\mathsf{dom}(D_{\text{Grohe}})$ is contained in the set of tuples of the form (v, e, i, p, z), for $v \in V$, $e \in E$, $i \in [k]$, p is a two-element subset of $[k]$, and $z \in \mathsf{dom}(D)$. Therefore, h_0 is well-defined on $\mathsf{dom}(D_{\text{Grohe}})$.
- Also, $D^* \subseteq D_{\text{Grohe}}$.

In addition, the following crucial property holds for D_{Grohe}.

Lemma H.1. ([26]) *Let h be an homomorphism from D to D_{Grohe} such that $h_0 \circ h$ is the identity. Then G contains a k-clique.*

Our variant still enjoys many of the good properties of Grohe's database, and some additional properties concerning the satisfaction of frontier-guarded TGDs, as shown in the following Theorem (which is essentially an equivalent reformulation of Theorem 7.1) for the case when CQs allow for free variables.

Lemma H.2. *Let G, k, D, D', A, $D^* = D^*(G, D, D', A, \mu)$, and h_0 be as defined above. The following statements hold:*

(1) There is a polynomial-time algorithm that, given G, D, D', A, and μ, computes D^*.
(2) h_0 is a surjective homomorphism from D^* to D'.
(3) G contains a k-clique iff there is a homomorphism h from D to D^* such that $h_0(h(\cdot))$ is the identity on A.
(4) If $D' \models \Sigma$, and every clique of size at most $3 \cdot r$ in G is contained in a clique of size $3 \cdot r \cdot m$, then $D^* \models \Sigma$.

Proof. Items (1) and (2) follow directly from the definition of D^* and h_0. However, for item (1) it is necessary to note that since the arity of the schema $\mathbf{S}$ is bounded by the fixed integer r, in the construction of D it is enough to consider only all labelled cliques of size at most r.

Let us prove (3). Although ($\Leftarrow$) can be obtained easily by mimicking the proof in [26], we shall give a proof using Lemma H.1. Assume that h is a homomorphism from D to D^* such that $h_0 \circ h$ is the identity on A. Then the restriction $h_{|A}$ of h to A defines a homomorphism from $D_{|A}$ to D^* such that $h_0 \circ h_{|A}$ is the identity. But then $h_{|A}$ corresponds to a homomorphism from $D_{|A}$ to $D(G, D_{|A}, D_{|A}, \mathsf{dom}(D_{|A}), \mu)$ with $h_0 \circ h_{|A}$ being the identity. Since $D(G, D_{|A}, D_{|A}, A, \mu)$ is contained in the original Grohe's database, $D_{\text{Grohe}}(G, D_{|A}, \mu)$, it follows from Lemma H.1 that G contains a k-clique. For the converse ($\Rightarrow$) assume that G has a k-clique, or alternatively, that there is a labelled clique η in G with domain $[k]$. Then η covers all elements in $\mathsf{dom}(D')$ and, hence, the mapping $h : \mathsf{dom}(D) \to \mathsf{dom}(D^*)$ with $h(z) = z_\eta$, for every $z \in \mathsf{dom}(D)$, defines a homomorphism from D to D^* satisfying the conditions.

Let us finally show (4). Consider an arbitrary frontier-guarded TGD $\phi(\bar{x}, \bar{y}) \to \exists \bar{z} \psi(\bar{x}, \bar{z})$ in Σ, and let $(\bar{a}, \bar{b})$ be an arbitrary tuple in the evaluation of $\phi(\bar{x}, \bar{y})$ over D^*. Since h_0 is a homomorphism from D^* to D', we have that D' contains all facts in $\phi(h_0(\bar{a}, \bar{b}))$. Therefore, since $D' \models \Sigma$, there is a tuple $\bar{d}$ of elements in $\mathsf{dom}(D')$, for $|\bar{d}| = |\bar{z}|$, such that D' contains every atom in $\psi(h_0(\bar{a}), \bar{d})$. Since there is an atom guarding $\bar{x}$ in ϕ, it follows from the definition of D^* that $\bar{a} = h_0(\bar{a})_\eta$ for some labelled clique η covering $h_0(\bar{a})$. In addition, $\Sigma \in \mathbb{FG}_m$ and hence $(h_0(\bar{a}), \bar{d})$ can have at most $r \cdot m$ elements. It follows from the assumptions of item (4) that there is labelled clique η' in G, extending η, that covers all elements mentioned by $\bar{d}$ (in addition to all elements in $h_0(\bar{a})$ already covered by η). Thus, D^* contains every atom in $\psi(h_0(\bar{a})_{\eta'}, \bar{d}_{\eta'}) = \psi(\bar{a}, \bar{d}_{\eta'})$. □

H.2 A Crucial Lemma

Our proof borrows several ideas and techniques from the proof of Grohe's Theorem. However, the adaptation of such techniques is non-trivial for the reasons that we have already discussed in Section 5.2, and we briefly recall here again. First, we cannot construct an arbitrary database D, but we need to construct one that satisfies the given set Σ of frontier-guarded TGDs. Second, the notion of core of a CQ, which is crucial for the proof of Grohe, cannot be directly used in our context. Instead, we need to develop a technical lemma, which, intuitively speaking, states that some subsets of our CQs behave like cores for the sake of our proof.

Lemma H.3. *Fix $\ell \geq r \cdot m$. There is a computable function that takes as input a CQS $S = (\Sigma, q(\bar{x}))$ from $(\mathbb{FG}_m, \mathbb{CQ})$ that is* not *uniformly $\mathbb{CQ}_\ell$-equivalent, and a positive integer $s \geq 1$, and outputs a CQ $p(\bar{x})$, a subset X of the existentially quantified variables of p, and a CQ $p'(\bar{x})$, such that:*

(1) $q \equiv_\Sigma p$.
(2) $D[p'] \models \Sigma$.
(3) $D[p] \subseteq D[p']$.
(4) For every homomorphism h from p to p' with $h(\bar{x}) = \bar{x}$, we have that $h(X) = X$.
(5) The treewidth of $G^p_{|X}$ is larger than ℓ.
(6) For every CQ $p''(\bar{x})$ with at most s variables, if $\bar{x} \in p''(D[p'])$ then also $\bar{x} \in p''(\mathsf{chase}(D[p], \Sigma))$.

Proof. Let us assume that S is defined over schema $\mathbf{S}$, that $q(\bar{x})$ has $n > 0$ variables, and that t is the maximum number of variables appearing in the body of some TGD in Σ. We set $s' = \max\{s, n, t\}$.

First of all, we can compute by exhaustive search a CQ $p(\bar{x})$ with a minimum number of variables that satisfies $q \equiv_\Sigma p$. This is because the notion $\equiv_\Sigma$ is decidable, for $\Sigma \in \mathbb{FG}$. In fact, to check whether $p \subseteq_\Sigma q$ one can construct the finite instance $M = M(D[p], \Sigma, n)$ from Definition 6.5, which exists by the strong finite controllability of $\mathbb{FG}$ as stated in Theorem 6.7, and then check whether $\bar{x} \in q(M)$. We know that the latter holds iff $\bar{x} \in q(\mathsf{chase}(D[p], \Sigma))$. From Proposition 4.5 we get that $p \subseteq_\Sigma q$. Analogously, one can check $q \subseteq_\Sigma p$.

We have then that item (1) holds by definition. We now explain how to build p' and X. We need some preparation. First, we shall

assume that for every TGD σ in Σ the new TGD σ' obtained by identifying two variables in the body of σ belongs also to Σ. We can assume this without loss of generality because σ' is implied by σ and, furthermore, there is an easy procedure that iteratively adds to Σ all new TGDs that can be obtained by identifying two variables in the body. Also, we build a new relational schema $\mathbf{S}'$ and set Σ' of frontier-guarded TGDs in the following way. Initially, set $\mathbf{S}' = \mathbf{S}$ and $\Sigma' = \Sigma$. Then, for each $\sigma = \phi(\bar{x}', \bar{y}) \rightarrow \exists \bar{w} \psi(\bar{x}', \bar{w})$ in Σ, for every partition $\chi \wedge \chi' = \phi$ of the atoms in ϕ, and for every CQ $q'(\bar{z}) = \exists \bar{v} \chi(\bar{v}, \bar{z})$ obtained by quantifying existentially some variables in χ, we include in $\mathbf{S}'$ a new predicate $T_{q'}$ with the same arity than $\bar{z}$. Furthemore, for every atom $R(\bar{u})$ where $R \in \mathbf{S}$ and $\bar{u}$ might contain variables occurring in ϕ and new fresh variables, we include in Σ' two new TGDs σ' and σ'' (provided they are in $\mathbb{FG}$). TGD σ' is obtained by replacing, in ϕ, $\chi(\bar{v}, \bar{z})$ by $T_{q'}(\bar{z}) \wedge R(\bar{u})$, whereas σ'' is the TGD $\phi(\bar{v}, \bar{z}) \wedge R(\bar{u}) \rightarrow T_{q'}(\bar{z})$.

The following claim follows easily from the definition.

Claim H.4. *For every* $\mathbf{S}$*-instance* D*, and every atom* $R(\bar{c})$ *where* $R \in \mathbf{S}$ *and* $\bar{c}$ *only mentions elements from* $\mathsf{dom}(D)$*,*

$$R(\bar{c}) \in \mathsf{chase}(D, \Sigma) \iff R(\bar{c}) \in \mathsf{chase}(D, \Sigma').$$

We use $\mathcal{A}$ to denote the collection of all subsets A from $\mathsf{dom}(p)$ that are guarded, i.e., there exists an atom $R(\bar{a})$ of p such that $\bar{a}$ mentions all elements in A. For every $A \in \mathcal{A}$, let E_A be the $\mathbf{S}'$-database that contains all facts $R(\bar{b}) \in \mathsf{chase}(D[p], \Sigma')$ for which it is the case that all elements from $\bar{b}$ appear in A. Notice that, by Theorem 6.7, each E_A can be constructed from $D[p]$, Σ' and A. It follows also from Theorem 6.7 that we can construct a finite $\mathbf{S}'$-instance $M_A = M(E_A, \Sigma', s')$ with the property that, for every CQ $p''(\bar{x}'')$ with at most s' variables,

$$\bar{a} \in p''(M_A) \iff \bar{a} \in p''(\mathsf{chase}(E_A, \Sigma')),$$

for every tuple $\bar{a}$ of elements in $\mathsf{dom}(E_A)$ of the same arity as $\bar{x}''$.

Let us define D_A by removing from M_A those atoms that are not over $\mathbf{S}$. We can assume (renaming elements if necessary) that $\mathsf{dom}(D_A) \cap \mathsf{dom}(D_{A'}) = A \cap A'$, for every $A, A' \in \mathcal{A}$. We then define p' as $\cup_{A \in \mathcal{A}} D_A$. We have by construction that $D[p] \subseteq D[p']$, i.e., item (3) holds.

We now show item (6). We actually prove something stronger:

Claim H.5. *Let* $p''(\bar{x})$ *be a CQ with at most* s *variables and* $\bar{c} \in \mathsf{dom}(p)^{|\bar{x}|}$ *such that* $\bar{c} \in p''(D[p'])$*. Then* $\bar{c} \in p''(\mathsf{chase}(D[p], \Sigma))$*.*

Proof. Assume that $\bar{c} \in p''(D[p'])$, i.e., there exists a homomorphism h from p'' to p' with $h(\bar{x}) = \bar{c}$. Let $p''(\bar{x}) = \exists \bar{y} \phi(\bar{x}, \bar{y})$ and for every $A \in \mathcal{A}$, let ϕ_A be the conjunction of all atoms, $R(\bar{z})$, in ϕ such that $h(\bar{z})$ is entirely contained in $\mathsf{dom}(D_A)$. Notice that if we define p''_A as the CQ $\exists \bar{y}_A \phi_A(\bar{x}_A, \bar{y}_A)$, where $\bar{x}_A$ and $\bar{y}_A$ are all variables of ϕ_A occurring in $\bar{x}$ and $\bar{y}$, respectively, then $p'' \equiv \bigwedge_{A \in \mathcal{A}} p''_A$.

Since $h(\bar{x}_A) \in p''_A(D_A)$, we have by definition that also $h(\bar{x}_A) \in p''_A(M_A)$. Since p''_A has at most $s \leq s'$ variables, it follows then from the definition of M_A that $h(\bar{x}_A) \in p''_A(\mathsf{chase}(E_A, \Sigma'))$, and hence $h(\bar{x}_A) \in p''_A(\mathsf{chase}(D[p], \Sigma'))$. Since A was arbitrarily chosen, we can then conclude that $h(\bar{x}) \in p''(\mathsf{chase}(D[p], \Sigma'))$. Consequently, since p'' only contains predicates in $\mathbf{S}$ it follows directly from Claim H.4 that $h(\bar{x}) \in p''_A(\mathsf{chase}(D[p], \Sigma))$. □

We then show that item (2) holds.

Claim H.6. *It is the case that* $D[p'] \models \Sigma$*.*

Proof. Let $\sigma = \phi(\bar{x}', \bar{y}) \rightarrow \exists \bar{w} \psi(\bar{x}', \bar{w})$ be any TGD in Σ and let h be any mapping of the variables in ϕ to elements in $\mathsf{dom}(p')$ such that $\phi(h(\bar{x}'), h(\bar{y})) \subseteq p'$. Our goal is to show that $h(\bar{x})$ belongs to the evaluation of $\exists \bar{w} \psi(\bar{x}', \bar{w})$ in $D[p']$. We can assume that h is injective (no two variables from ϕ are mapped to the same element in $\mathsf{dom}(p')$) since the non-injective case is reduced to the injective one. Indeed, if h is non-injective we only need to replace σ by the TGD obtained by identifying variables in the body that are mapped by h to the same element (which by assumption also belongs to Σ') and also modify h accordingly.

Since $\Sigma \in \mathbb{FG}$, it follows that there exists some $A \in \mathcal{A}$ such that $h(\bar{x}')$ is entirely contained in $\mathsf{dom}(D_A)$. If $h(\bar{y})$ is also contained in $\mathsf{dom}(D_A)$ then there is nothing else to prove since, by construction, D_A satisfies Σ.

Assume, then, that $h(\bar{y})$ is not contained in $\mathsf{dom}(D_A)$. Let $R(\bar{a})$ be the atom in p that guards A. Let $\bar{a} = (a_1, \ldots, a_s)$ and let $\bar{u} = (u_1, \ldots, u_s)$ be a tuple of variables where for every $1 \leq i \leq s$, u_i is a new fresh variable if a_i does not belong to the image of h, and $u_i = h^{-1}(a_i)$ otherwise (note that here we are using the fact that h is injective).

Let $\bar{y}'$ be a tuple containing all variables in $\bar{y}$ that are not mapped by h inside $\mathsf{dom}(D_A)$, let $\chi(\bar{y}', \bar{z})$ be the conjunction of all atoms in ϕ containing some variable in $\bar{y}'$ (where $\bar{z}$ is a tuple that contains the rest of the variables in χ), and let $q'(\bar{z}) = \exists \bar{y}' \chi(\bar{y}', \bar{z})$. Note that $h(\bar{z}) \subseteq A$ as, by construction, if a tuple in p' contains elements in $\mathsf{dom}(D_A) \setminus A$ then all of its elements must necessarily be contained in A. Since every element mentioned in $h(\bar{z})$ belongs to A, it follows that the TGD $\sigma' = \chi(\bar{z}, \bar{y}) \wedge R(\bar{u}) \rightarrow R_{q'}(\bar{z})$ is frontier-guared by $R(\bar{u})$ and, consequently, it belongs to Σ'. Let h' be the extension of h so that every fresh variable u_i in $\bar{u}$ is mapped by h' to a_i. Note that the image of h' is contained in $\mathsf{dom}(p')$ and hence it follows, by applying TGD σ with mapping h', that $R_{q'}(h(\bar{z})) \in (\mathsf{chase}(D[p], \Sigma'))$. Consequently, $R_{q'}(h(\bar{z}))$ belonts to E_A and, hence, to M_A, as well.

Consider the TGD σ'' obtained by replacing $\chi(\bar{y}', \bar{z})$ by $R_{q'}(\bar{z}) \wedge R(\bar{u})$ in the body, $\phi(\bar{x}', \bar{y})$, of σ. That is, $\sigma'' = \chi'(\bar{x}', \bar{z}) \wedge R_{q'}(\bar{z}) \wedge R(\bar{u}) \rightarrow \exists \bar{w} \psi(\bar{x}', \bar{w})$, where χ' is the set of all atoms in ϕ that are not in χ. We shall prove that σ'' is frontier-guarded (and hence, it belongs to Σ'). Clearly $\bar{x}'$ is guarded by some atom in χ or χ'. In the latter case there is nothing else to prove so we can assume that χ contains some atom guarding $\bar{x}'$. By definition, this atom contains some elements not in D_A which implies that $\bar{x}'$ is entirely contained in A. Consequently $\bar{x}'$ is guarded by $R(\bar{u})$.

Now, since $R_{q'}(h(\bar{z})) \in M_A$ it follows, by applying σ'' with mapping h', that $h(\bar{x}')$ belongs to the evaluation of $\exists \bar{w} \psi(\bar{x}', \bar{w})$ over M_A. Since all atoms in ψ involve only predicates from $\mathbf{S}$, it follows that $h(\bar{x}')$ belongs to the evaluation of $\exists \bar{w} \psi(\bar{x}', \bar{w})$ over D_A, and hence over $D[p']$. □

Before continuing, we prove the following important properties of the homomorphisms from p to p'.

Claim H.7. *Let* h *be any homomorphism from* p *to* p' *with* $h(\bar{x}) = \bar{x}$*. The following statements hold:*

(a) It is the case that $p \equiv_\Sigma p'_{|\mathsf{dom}(h(p))}$*.*

(b) *h is injective.*
(c) *For every $A \in \mathcal{A}$, it is the case that $(D_A)_{|\mathsf{dom}(h(p))\cap\mathsf{dom}(D_A)}$ has treewidth at most $r \cdot m$.*
(d) *For every $Y \subseteq \mathsf{dom}(p) \setminus \bar{x}$, the treewidth of $G^p_{|Y}$ is at most the maximum between $r \cdot m$ and the treewidth of $G^p_{|h(Y)\cap\mathsf{dom}(p)}$.*

Proof. Consider first item (a). Clearly $p'_{|\mathsf{dom}(h(p))} \subseteq_\Sigma p$ since h defines a homomorphism from p to $p'_{|\mathsf{dom}(h(p))}$. For the converse, we have $\bar{x} \in p'(D[p'])$ and, therefore, $\bar{x} \in p'_{|\mathsf{dom}(h(p))}(D[p'])$. Since p, and thus $p'_{|\mathsf{dom}(h(p))}$, has at most has at most $n \leq s'$ variables, it follows from Claim H.5 that $\bar{x} \in p'(\mathsf{chase}(D[p], \Sigma))$, i.e., $p \subseteq_\Sigma p'_{|\mathsf{dom}(h(p))}$. Item (b) holds due to the fact that $p \equiv_\Sigma p'_{|\mathsf{dom}(h(p))}$ and the minimality condition we have imposed in the definition of p.

Let us prove item (c). Let $A \in \mathcal{A}$. Since $h(p)$ is of size at most $n \leq s'$, it follows directly from the construction of D_A that there exists a homomorphism g_A from $(D_A)_{|\mathsf{dom}(h(p))\cap\mathsf{dom}(D_A)}$ to $\mathsf{chase}(E_A, \Sigma')$ that acts as the identity on $A \cap \mathsf{dom}(h(p))$ (simply take the CQ induced in D_A by $h(p)$ with all variables corresponding to elements in $A \cap \mathsf{dom}(h(p))$ being free). Let us denote by F_A the subinstance of $\mathsf{chase}(E_A, \Sigma')$ that contains only those atoms $R(\bar{a})$, for $R \in \mathbf{S}$, such that all the elements mentioned in $\bar{a}$ belong to the image of g_A. We show next that there is a homomorphism f_A from F_A to $\mathsf{chase}(p, \Sigma)$ that acts as the identify on A.

Let $p_A(\bar{x}_A)$ be a CQ defined as the conjunction of atoms in F_A, where every constant c is replaced by a variable x_c and $\bar{x}_A$ is the tuple of variables that represent the elements $\bar{a}$ in $\mathsf{dom}(F_A) \cap A$. It is then the case that $\bar{a} \in p_A(\mathsf{chase}(E_A, \Sigma'))$. By definition of E_A, this means that $\bar{a} \in p_A(\mathsf{chase}(p, \Sigma'))$. But p_A only mentions relation symbols in $\mathbf{S}$, and hence it is the case that $\bar{a} \in p_A(\mathsf{chase}(p, \Sigma))$ from Claim H.4. This implies our claim for the existence of f_A as described above.

Consequently, the mapping f that sends every element a in $\mathsf{dom}(p)$ to $f_A \circ g_A \circ h(a)$, where A is any set in $\mathcal{A}$ with $a \in A$, is well-defined and corresponds to a homomorphism from p to $\mathsf{chase}(p, \Sigma)$. Clearly, $\mathsf{chase}(p, \Sigma)_{|\mathsf{dom}(f(p))} \equiv_\Sigma p$, and, hence, by the minimality of p it follows that f is injective.

Let $A \in \mathcal{A}$. Since $|\mathsf{dom}(E_A)| \leq r$ and $\Sigma' \in \mathbb{FG}_m$, it follows directly that $\mathsf{chase}(E_A, \Sigma')$, and thus F_A, has a treewidth at most $r \cdot m$. Since f is injective, then g_A is also injective. Therefore, $(D_A)_{|\mathsf{dom}(h(p))\cap\mathsf{dom}(D_A)}$, which is the preimage of g_A, also has treewidth at most $r \cdot m$. This establishes item (c).

We now prove item (d). Let (T, χ) be a tree decomposition of $G^{p'}_{|h(Y)\cap\mathsf{dom}(p)}$. We shall extend it into a tree decomposition of $G^{p'}_{|h(Y)}$ in the following way. Let $A \in \mathcal{A}$. We know from item (c) that $(D_A)_{|\mathsf{dom}(h(p))\cap\mathsf{dom}(D_A)}$ has treewidth at most $r \cdot m$. This implies

$$H_A := G^{p'}_{|h(Y)\cap\mathsf{dom}(D_A)}$$

has a tree decomposition (T_A, χ_A) of width at most $r \cdot m$ (as A is guarded, and thus every pair of elements in $G^{p'} \cap A$ is linked by an edge). Moreover, since A is a guarded in $\mathsf{dom}(p)$, the elements in $A \cap h(Y)$ define a clique on H_A. Hence, there is a node $v \in T_A$ such that $A \cap h(Y) \subseteq \chi_A(v)$. Likewise, there is $u \in T$ such that $A \cap h(Y) \subseteq \chi(u)$. Thus, we can construct a new tree decomposition by joining u and v with an edge. Iterating this process for all $A \in \mathcal{A}$ we get a tree decomposition of $G^{p'}_{|h(Y)}$ whose width is the maximum between $r \cdot m$ and the width of T. Since h is injective by item (b), $G^p_{|Y}$ has a tree decomposition of the same width as $G^{p'}_{|h(Y)}$. □

Let k be the treewidth of G^p. Since $q \equiv_\Sigma p$ and q is not uniformly $\mathbb{CQ}_\ell$-equivalent, it follows that $k > \ell$. A set $Y \subseteq \mathsf{dom}(p) \setminus \bar{x}$ is *nice* if $G^p_{|Y}$ has treewidth k and $|Y|$ is minimal with such property. We construct X as the set containing all elements that belong to some nice set. It follows directly from the definition that property (5) holds. To finish, we show that property (4) is satisfied.

Claim H.8. *Let h be a homomorphism from p to p' with $h(\bar{x}) = \bar{x}$. Then $h(X) = X$*

Proof. Let Y be any nice set. From Claim H.7(c), the treewidth of $G^p_{|Y}$ is at most the maximum between r, m, and the treewidth of $G^p_{|h(Y)\cap\mathsf{dom}(p)}$. Since G^p_Y has treewidth k and $k > \ell \geq r \cdot m$, the treewidth of $G^p_{|h(Y)\cap\mathsf{dom}(p)}$ is at least k, implying that it is, in fact, k since the treewidth of G^p is k. By the minimality of $|Y|$, it must be the case then that $h(Y) \subseteq \mathsf{dom}(p)$. Therefore, $h(Y)$ is also a nice set and, hence, $h(Y) \subseteq X$. Since h is injective from Claim H.7(b), it must be the case that $h(X) = X$. □

This finishes the proof of Lemma H.3. □

We now proceed to explain how Lemma H.3 is applied in order to prove Theorem 5.13.

H.3 The FPT-reduction

We have all the ingredients needed to define the fpt-reduction used in the proof of Theorem 5.13. Let (G, k) be an instance of p-Clique. It is easy to see that we can assume, without loss of generality, that every clique of size at most $3 \cdot r$ in G is contained in a clique of size $3 \cdot r \cdot m$; recall that r is the maximum arity of the predicates occurring in CQSs of $\mathbb{O}$, while m is the maximum number of atoms in the head of the TGDs occurring in CQSs of $\mathbb{O}$.

Claim H.9. p-Clique *is* W[1]*-hard, even if restricted to instances that satisfy the above restriction.*

Henceforth, for $k \geq 1$ we let $K = \binom{k}{2}$. From the Excluded Grid Theorem [34], there is a computable function $F : \mathbb{N} \to \mathbb{N}$ such that, for each $k \geq 1$ and simple graph G of treewidth at least $F(k)$, we have that at least some connected component of G contains a $(k \times K)$-grid as a minor. By hypothesis on the class $\mathbb{O}$, there exists a CQS $S = (\Sigma, q'(\bar{x}))$ from $\mathbb{O}$ such that $S \notin (\mathbb{FG}_m, \mathbb{UCQ})^{\equiv}_{F(k)}$, where F is as defined above. W.l.o.g., we assume that $F(k) \geq r \cdot m$. We build from (G, k) an instance $(D^*, \Sigma, q'(\bar{x}))$ of p-CQS-Evaluation($\mathbb{O}$), where D^* is a database defined as follows.

First, observe that there must be a CQ $q(\bar{x})$ in $q'(\bar{x})$ such that $(\Sigma, q(\bar{x})) \notin (\mathbb{FG}_m, \mathbb{CQ})^{\equiv}_{F(k)}$, while $q \not\subseteq_\Sigma \hat{q}$ for every disjunct $\hat{q}$ of q' other than q. In fact, if we remove from q' every disjunct that is not maximal with respect to $\subseteq_\Sigma$, then we obtain an equivalent UCQ $q''(\bar{x})$ under Σ. Therefore, $(\Sigma, q''(\bar{x})) \notin (\mathbb{FG}_m, \mathbb{UCQ})^{\equiv}_{F(k)}$, and hence, there is at least one disjunct q in q'' such that $(\Sigma, q(\bar{x})) \notin (\mathbb{FG}_m, \mathbb{CQ})^{\equiv}_{F(k)}$.

Since, by assumption, $F(k) \geq r \cdot m$, it is possible to compute from $q(\bar{x})$ and s, where s is the maximum number of variables over all CQs of q', a CQ $p(\bar{x})$, a subset X of the existentially quantified variables of p, and a CQ $p'(\bar{x})$, that satisfy the properties stated in Lemma H.3 for $\ell = F(k)$. In particular, the treewidth of $G^{p}_{|X}$ is at least $F(k)$, and hence, by the Excluded Grid Theorem, there exists a connected component H of $G^{p}_{|X}$ and a minor map μ from the $(k \times K)$-grid onto H. We then define D^* as $D^*(G, D[p], D[p'], X, \mu)$, where $D^*(G, D[p], D[p'], X, \mu)$ is our modified version of Grohe's database that satisfies the properties stated in Lemma H.2.

H.4 Correctness of the Reduction

It remains to show that the above is an fpt-reduction from p-Clique to CQS-Evaluation($\mathbb{O}$). To this end, we need to show the following.

Lemma H.10. *The following statements hold:*

(1) $D^* \models \Sigma$.
(2) G has a k-clique iff $\bar{x} \in q'(D^*)$.
(3) There are computable functions $f, g : \mathbb{N} \to \mathbb{N}$ *such that* $(D^*, \Sigma, q'(\bar{x}))$ *can be constructed in time* $f(k) \cdot ||G||^{O(1)}$ *and* $(||q'|| + ||\Sigma||) \leq g(k)$.

Proof. The proof of (1) follows from the last item in Theorem H.2 since $D[p'] \models \Sigma$ and every clique of size at most $3 \cdot r$ in G is contained in a clique of size $3 \cdot r \cdot m$. It is worth noticing though that this holds for our variant of Grohe's database, but not necessarily for the one that Grohe originally defined in [26].

We now proceed to show item (2).

($\Rightarrow$) Assume that G has a k-clique. Then, by Lemma H.2, there is a homomorphism h from p to D^* such that $h_0 \circ h$ is the identity on X. It follows directly from the definition of D^* that the mapping h' defined as $h'(x) = h(x)$, when $x \in X$, and $h'(x) = x$, otherwise, is also a homomorphism from p to D^*. In particular, since X contains only quantified variables we have that $h'(\bar{x}) = \bar{x}$, and hence $\bar{x} \in p(D^*)$. But $q \equiv_\Sigma p$, and thus, $\bar{x} \in q(D^*)$ since, by item (1), $D^* \models \Sigma$. Therefore, $\bar{x} \in q'(D^*)$ as q is a disjunct of q'.

($\Leftarrow$) Conversely, assume that $\bar{x} \in q'(D^*)$. First observe that $\bar{x} \in q(D^*)$. By contradiction, assume otherwise. Then, there is a CQ $\hat{q}(\bar{x})$ in $q'(\bar{x})$, other than q, such that $\bar{x} \in \hat{q}(D^*)$. Clearly, D^* homomorphically maps to p' via the mapping h_0, and hence, there is a homomorphism from $\hat{q}$ to p' mapping $\bar{x}$ to $\bar{x}$. By Lemma H.3, $\bar{x} \in \hat{q}(\mathsf{chase}(p, \Sigma))$, which in turn implies that $p \subseteq_\Sigma \hat{q}$; the latter holds due to Proposition 4.5. Hence, $q \subseteq_\Sigma \hat{q}$ as $p \equiv_\Sigma q$, which contradicts the way in which q has been chosen.

Now, since $\bar{x} \in q(D^*)$, we have that $\bar{x} \in p(D^*)$ because $q \equiv_\Sigma p$ and $D^* \models \Sigma$. Then, there is a homomorphism h from p to D^* with $h(\bar{x}) = \bar{x}$. It follows that $h_0 \circ h$ is a homomorphism from p to p' that maps $\bar{x}$ to $\bar{x}$. In consequence, from Lemma H.3 we obtain that $h(X) = X$. Therefore, there must exist some $m \geq 0$ such that $g = h \circ (h_0 \circ h)^m$ is a homomorphism from p to D^* that satisfies that $h_0 \circ g$ is the identity on X. It follows from Lemma H.2 that G has a k-clique.

As for item (3), first notice that the CQS $S = (\Sigma, q')$ can be computed by simply enumerating the CQSs from $\mathbb{O}$ until we find S since, by Theorem 5.10, we can check whether $S \notin (\mathbb{FG}, \mathbb{UCQ})^{\equiv}_{F(k)}$. The same holds for q. From q we can construct the CQs p and p', as well as the set of variables X, by applying Lemma H.3. We can then compute μ via an exhaustive search over $G^{p'}$. Notice that the construction of q', q, p, p', X, and μ depends only on k. Lemma H.2 states, on the other hand, that it is possible to construct D^* in time polynomial, given p', p, X, μ, and G. Putting all these together, we obtain the existence of computable functions $f, g : \mathbb{N} \to \mathbb{N}$ as required in item (3), and the claim follows. □

I PROOF OF LEMMA 6.6

For brevity, let $\mathsf{finans}(q, D, \Sigma) = \bigcap_{M \in \mathsf{fmods}(D,\Sigma)} q(M)$.

($\Rightarrow$) Consider an S-database D, a set $\Sigma \in \mathbb{C}$ of TGDs over S, and an integer $n \geq 0$. Assuming that $\mathsf{dom}(D) = \{d_1, \ldots, d_\ell\}$, for $\ell \geq 1$, let $D^+ = D \cup \{\mathrm{Dom}(d_1), \ldots, \mathrm{Dom}(d_\ell)\}$, and $\bar{d} = (d_1, \ldots, d_\ell)$.

Lemma I.1. *There exists an instance* $M^* \in \mathsf{fmods}(D^+, \Sigma)$ *such that, for every UCQ* q^* *over* $\mathbf{S} \cup \{\mathrm{Dom}\}$ *of arity* ℓ *with at most* $n + \ell$ *variables,* $\bar{d} \in q^*(M^*)$ *implies* $\bar{d} \in q^*(\mathsf{chase}(D^+, \Sigma))$

Proof. Let q be the UCQ of arity ℓ consisting of all the CQs q' with at most $n + \ell$ variables such that $\bar{d} \notin q'(\mathsf{chase}(D^+, \Sigma))$. Clearly, $\bar{d} \notin q(\mathsf{chase}(D^+, \Sigma))$. Since $\mathbb{C}$ is finitely controllable, we conclude that $\bar{d} \notin \mathsf{finans}(q, D^+, \Sigma)$. Thus, there exists an instance $M^* \in \mathsf{fmods}(D^+, \Sigma)$ such that $\bar{d} \notin q(M^*)$. Fix an arbitrary UCQ q^* of arity ℓ with at most $n + \ell$ variables. Observe that $\bar{d} \notin q^*(\mathsf{chase}(D^+, \Sigma))$ implies $\bar{d} \notin q^*(M^*)$ since, by construction, each CQ in q^* occurs also in q, and we know that $\bar{d} \notin q(M^*)$. The claim follows. □

We claim that the desired finite model $M(D, \Sigma, n)$ is the instance $M^*_{|\mathbf{S}}$, where M^* is the $(\mathbf{S} \cup \{\mathrm{Dom}\})$-instance of $\mathsf{fmods}(D^+, \Sigma)$ provided by Lemma I.1. To this end, we need to show that, for every UCQ $q(\bar{x})$ over S of arity $r \geq 0$ with at most n variables, $q(\mathsf{chase}(D, \Sigma)) = q(M^*_{|\mathbf{S}})$. Fix an arbitrary tuple $\bar{c} \in \mathsf{dom}(D)^r$. Clearly, $\bar{c} \in q(\mathsf{chase}(D, \Sigma))$ implies $\bar{c} \in q(M^*_{|\mathbf{S}})$ since $M^*_{|\mathbf{S}}$ is a model of D and Σ. Assume now that $\bar{c} \in q(M^*_{|\mathbf{S}})$. We proceed to show that $\bar{c} \in q(\mathsf{chase}(D, \Sigma))$. It is not difficult to see that there exists a UCQ $\hat{q}$ of arity ℓ over $(\mathbf{S} \cup \{\mathrm{Dom}\})$ with at most $n + \ell$ variables such that:

$$\bar{c} \in q(M^*_{|\mathbf{S}}) \implies \bar{d} \in \hat{q}(M^*) \tag{1}$$

$$\bar{d} \in \hat{q}(\mathsf{chase}(D^+, \Sigma)) \implies \bar{c} \in q(\mathsf{chase}(D, \Sigma)) \tag{2}$$

Since $\hat{q}$ has at most $n + \ell$ variables, by Lemma I.1, we get that

$$\bar{d} \in \hat{q}(M^*) \implies \bar{d} \in q(\mathsf{chase}(D^+, \Sigma)) \tag{3}$$

Since, by hypothesis, $\bar{c} \in q(M^*_{|\mathbf{S}})$, from (1) we get that $\bar{d} \in \hat{q}(M^*)$. Thus, by (3), we get that $\bar{d} \in q(\mathsf{chase}(D^+, \Sigma))$, and hence, by (2), we conclude that $\bar{c} \in q(\mathsf{chase}(D, \Sigma))$, as needed.

($\Leftarrow$) Consider an S-database D, a set $\Sigma \in \mathbb{C}$ of TGDs over S, and a UCQ $q(\bar{x})$ over S. Fix an arbitrary tuple $\bar{c} \in \mathsf{dom}(D)^{|\bar{x}|}$. We need to show that $\bar{c} \in q(\mathsf{chase}(D, \Sigma))$ iff $\bar{c} \in \mathsf{finans}(q, D, \Sigma)$. Clearly, the ($\Rightarrow$) direction holds trivially. It remains to show that $\bar{c} \notin q(\mathsf{chase}(D, \Sigma))$ implies $\bar{c} \notin \mathsf{finans}(q, D, \Sigma)$. Let $n \geq 0$ be the number of variables occurring in q. Since $\mathbb{C}$ is strongly finitely controllable, there exists $M(D, \Sigma, n) \in \mathsf{fmods}(D, \Sigma)$ such that $\bar{c} \notin q(M(D, \Sigma, n))$. This immediately implies that $\bar{c} \notin \mathsf{finans}(q, D, \Sigma)$.

J PROOF OF THEOREM 6.7

To show that $\mathbb{FG}$ is strongly finitely controllable, by Lemma 6.6, it suffices to show that it is finitely controllable. This can be easily shown by exploiting the fact that the *guarded negation fragment of first-order logic* (GNFO) [5] enjoys the finite model property, i.e., if a GNFO sentence has a model, then it has a finite one. But let us first recall the guarded negation fragment.

GNFO restricts first-order logic by requiring that all occurrences of negation are of the form $\alpha \wedge \neg\varphi$, where α is an atom containing all the free variables of φ. Formally, the formulas of GNFO are generated by the recursive definition

$$\varphi \ ::= \ R(t_1, \ldots, t_n) \mid t_1 = t_2 \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \exists x\, \varphi \mid \alpha \wedge \neg\varphi,$$

where each t_i is a term (constant or variable), and in the last clause, α is an atomic formula containing all free variables of φ. We know that GNFO enjoys the finite model property, and we also have an upper bound on the size of finite models:

Proposition J.1 ([5]). *Consider a GNFO sentence φ. If φ has a model, then it has a finite one of size $2^{2^{||\varphi||^{O(1)}}}$.*

We proceed to show that $\mathbb{FG}$ is finitely controllable. Consider a database D, a set $\Sigma \in \mathbb{FG}$, and a UCQ $q(\bar{x})$. We need to show that

$$q(\mathsf{chase}(D, \Sigma)) \ = \bigcap_{M \in \mathsf{fmods}(D,\Sigma)} q(M).$$

It should be clear that the ($\subseteq$) direction holds trivially. It remains to show the ($\supseteq$) direction. Consider a tuple $\bar{c} \in \mathsf{dom}(D)^{|\bar{x}|}$. Our goal is to devise a GNFO sentence Φ such that

(1) $\bar{c} \in \bigcap_{M \in \mathsf{fmods}(D,\Sigma)} q(M) \implies \Phi$ does not have a *finite* model.
(2) Φ does not have a model $\implies \bar{c} \in q(\mathsf{chase}(D, \Sigma))$.

Having such a sentence in place we get the ($\supseteq$) direction since, by Proposition J.1, we can conclude that if Φ does not have a finite model, then it does not have a model at all, which in turn implies that $\bar{c} \in q(\mathsf{chase}(D, \Sigma))$, as needed. Let us now explain how Φ is constructed. We first observe that a frontier-guarded TGD σ of the form $\varphi(\bar{x}, \bar{y}) \rightarrow \exists \bar{x}\, \psi(\bar{x}, \bar{z})$ can be equivalently rewritten as

$$\phi_\sigma \ = \ \neg\Big(\exists \bar{x} \exists \bar{y}\big(\varphi(\bar{x}, \bar{y}) \wedge \neg\exists \bar{z}\, \psi(\bar{x}, \bar{z})\big)\Big),$$

which is a GNFO sentence since all the free variables of $\exists \bar{z}\, \psi(\bar{x}, \bar{z})$ occur in the guard of σ. We also observe that the sentence $\neg q(\bar{c})$ is trivially a GNFO sentence. Therefore,

$$\Phi \ = \ D \ \wedge \bigwedge_{\sigma \in \Sigma} \phi_\sigma \ \wedge \ \neg q(\bar{c}),$$

which is clearly the desired GNFO sentence.

It remains to show that a finite witness is realizable, i.e., there is a computable function that takes as input an $\mathbf{S}$-database D, a set $\Sigma \in \mathbb{FG}$ over $\mathbf{S}$, and an integer $n \geq 0$, and outputs the finite model $M(D, \Sigma, n)$. In fact, this follows from the proof of Lemma 6.6 and Proposition J.1. In the proof of Lemma 6.6, we actually show that $M(D, \Sigma, n)$ is the model of a GNFO sentence φ, and thus, due to Proposition J.1, we can assume that is of size $2^{2^{||\varphi||^{O(1)}}}$. Therefore, we can compute the finite model $M(D, \Sigma, n)$ as follows:

(1) we first compute the set S_n of all the non-isomorphic UCQs over $\mathbf{S}$ with at most n variables, which is clearly finite; and
(2) we enumerate all the non-isomorphic finite models M of φ of size $2^{2^{||\varphi||^{O(1)}}}$, which are finitely many, until we find one such that, for every $q \in S_n$, $q(\mathsf{chase}(D, \Sigma)) = q(M)$. Note that the latter equality can be effectively checked since $\Sigma \in \mathbb{FG}$.